

Universidad Carlos III de Madrid

Escuela Politécnica Superior

Ingeniería Técnica Industrial  
Electrónica Industrial



Proyecto Fin de Carrera

**DISEÑO Y PUESTA EN  
FUNCIONAMIENTO DE UN BRAZO  
ROBÓTICO IMPRIMIBLE**

**Autor:** Antonio Castro Gómez

**Tutor:** Alberto Valero Gómez

Leganés, Septiembre 2012



**Título:** Diseño y puesta en funcionamiento de un brazo robótico imprimible

**Asunto:** Memoria del Proyecto Fin de Carrera de Ingeniería Técnica industrial Electrónica

**Autor:** Antonio Castro Gómez

**Tutor:** Alberto Valero Gómez

Universidad Carlos III de Madrid

Campus de Leganés



# PROYECTO FIN DE CARRERA

Departamento de Ingeniería de Sistemas y Automática

Universidad Carlos III de Madrid

**Título:** Diseño y puesta en funcionamiento de un brazo robótico imprimible

**Autor:** Antonio Castro Gómez

**Tutor:** Alberto Valero Gómez

La lectura y defensa del presente Proyecto Fin de Carrera se realizó el día 21 de Septiembre de 2012 bajo el Tribunal:

- **Presidente:** Prof. D. Jose María Armingol. (UC3M)
- **Secretario:** Ing. D. José Medina Hernández. (UC3M)
- **Vocal:** Prof. D. Miguel Hernando (UPM)

Habiendo obtenido la calificación de:

**Presidente**

**Secretario**

**Vocal**

## Resumen

Este proyecto corresponde a un conjunto de trabajos del departamento de Ingeniería de Sistemas y Automática donde el principal objetivo es comprobar la viabilidad de los robots imprimibles, en este caso se trata de diseñar un brazo robótico educativo de bajo coste servocontrolado y con control en posición.

Para el diseño no hay unas pautas específicas, por lo que la imaginación del estudiante tiene un gran reto por delante, pues tendrá que diseñar todos los eslabones, de manera que sean imprimibles y que permita el movimiento relativo entre ellos, siempre usando programas de licencia libre.

En cuanto a los servomotores utilizados, se ha dejado a juicio del estudiante el tipo de servomotor a utilizar, pero teniendo siempre en mente que debe ser un brazo de bajo coste, por ello debe ingeniárselas para conseguir los objetivos con el menor precio posible.

Otro de los objetivos a cumplir es la manera en que se debe controlar el brazo, dicho control debe hacerse por el puerto serie (USB) de un PC, para ello se utilizará Arduino UNO creándose un programa que permita dicho control mediante una serie de comandos, cuyo formato se deja nuevamente a juicio del proyectante.

Finalmente una vez terminado el proyecto, se ha de realizar una secuencia completa del robot manipulando una pieza, comprobando que verdaderamente funciona. En este caso, la secuencia se encuentra en el siguiente enlace:

<http://www.youtube.com/watch?v=JnvWSHTnrqo&feature=plcp>

Por último, al tratarse de un brazo de licencia libre, realizado con impresoras 3D open source y con software libre, deberá ser subido íntegramente a la web de la comunidad de las impresoras 3D: [www.thingiverse.com](http://www.thingiverse.com), el enlace correspondiente al robot en dicha web es el siguiente:

<http://www.thingiverse.com/thing:30163>



# Índice general

## Tabla de contenido

Resumen .....	4
Índice general .....	6
Índice de figuras.....	8
Índice de tablas.....	10
Índice de ecuaciones .....	11
Capítulo 1: Introducción y Objetivos .....	13
1.1.    Introducción.....	13
1.2.    Prototipos de brazos robóticos para docencia, existentes en el mercado y objetivos a lograr.....	14
1.3.    Fases de desarrollo:.....	16
1.4.    Medios empleados: .....	18
Capítulo 2: Diseño mecánico .....	21
2.1.    Componentes imprimibles.....	21
2.1.1. Eslabones y movimiento relativo entre ellos.....	21
2.2.    Componentes no imprimibles.....	26
2.2.1.    Material de soporte complementario.....	26
2.2.2.    Servomotores modificados.....	28
Capítulo 3: Diseño electrónico y control.....	32
3.1.    Arduino UNO .....	32
3.2.    Electrónica complementaria (componentes característicos) .....	33
3.3.    Programación y funcionamiento (señales de control) .....	37
Capítulo 4: Montaje y cableado .....	43
4.1.    Montaje mecánico y cableado en la estructura.....	43
4.1.1.    Eslabón 0 .....	44
4.1.2.    Eslabón 1 .....	45
4.1.3.    Eslabón 2 .....	46
4.1.4.    Eslabón 3 .....	47
4.1.5.    Eslabón 4 .....	48
4.1.6.    Eslabón 5 y 6 .....	49



---

4.1.7.	Pinza .....	50
4.1.8.	Cableado.....	51
4.2.	Montaje electrónico y conexionado.....	57
4.2.1.	Montaje placa con electrónica complementaria .....	57
4.2.2.	Conexionado de placa complementaria y conexionado del cableado del brazo con Arduino UNO.....	58
4.3.	Puesta a punto .....	60
Capítulo 5: Modelo Cinemático.....		64
5.1.	Cinemática .....	64
5.2.	Cinemática Directa.....	65
5.3.	Cinemática Inversa .....	68
5.3.1.	Cinemática inversa para el posicionamiento .....	68
Capítulo 6: Conclusiones .....		73
6.1.	Conclusiones .....	73
REFERENCIAS .....		76
ANEXO I.....		78
A. Tiempo de impresión.....		79
B. Presupuesto.....		81
ANEXO II.....		82
Proyecto de J. Carlos Rodríguez Zambrana (Breve resumen. Objetivos).....		83

# Índice de figuras

Figura 1 - Gráfico de las aplicaciones de los Robots Industriales.....	13
Figura 2 - Gráfico de tiempo aproximado empleado por fase de desarrollo.....	16
Figura 3 - Eslabón 0 montado sin servos (Base del brazo robótico) .....	21
Figura 4 - Eslabón 1 montado sin servos.....	22
Figura 5 - Eslabón 2 montado sin servos.....	23
Figura 6 - Eslabón 3 montado sin servos.....	24
Figura 7 - Eslabón 4 montado sin servos.....	24
Figura 8 - Eslabón 5 montado sin servos.....	25
Figura 9 - Pinza montada sin servo .....	26
Figura 10 - Prolongador de servos .....	27
Figura 11 - Tornillería con tuercas y arandelas .....	28
Figura 12 - Diferencias entre la placa de un microservo y un servo estándar .....	30
Figura 13 - Arduino UNO R3 .....	32
Figura 14 - Esquemático de la placa para electrónica complementaria en Orcad .....	35
Figura 15 - Diseño de placa para electrónica complementaria en Orcad Layout .....	35
Figura 16 - Placa diseñada implementada y completa con componentes soldados .....	36
Figura 17 - Placas conectadas listas para el cableado.....	36
Figura 18 - Flujograma de la lectura estimada con un error de tres grados.....	38
Figura 19 - Flujograma de la lectura precisa a partir de la estimada con un error de tres grados .....	39
Figura 20 – Flujograma del programa principal .....	40
Figura 21 - Montaje de la Base.....	44
Figura 22 - Montaje eslabón 1. ....	45
Figura 23 - Montaje eslabón 2. ....	46
Figura 24 - Montaje eslabón 3. ....	47
Figura 25 - Montaje eslabón 4. ....	48
Figura 26 - Montaje eslabón 5. ....	49
Figura 27 - Montaje de la pinza.....	50
Figura 28 - Perfil izquierdo, parte frontal, parte trasera, perfil derecho del brazo. ....	51
Figura 29 - “Destape” eslabón 3 (análogo para eslabón 2).....	52
Figura 30 - Ángulo formado entre eslabón dos y tres en posiciones extremas.....	53
Figura 31 - Colocación de cables servos A1, C (los del interior) y alimentación (los del exterior con tubo termoretráctil negro). ....	54
Figura 32 - Colocación de cables en la base I. ....	55
Figura 33 - Colocación de cables en la base II. ....	55
Figura 34 - conectores soldados y brazo robótico cableado.....	57
Figura 35 - Placa de electrónica complementaria.....	57
Figura 36 - Diseño de placa de electrónica complementaria.....	58
Figura 37 - conexión de placa complementaria y Arduino UNO.....	58



Figura 38 - conectores y placa de electrónica complementaria. ....	59
Figura 39 - Placas conectadas al brazo, al puerto serie (USB) y a la fuente de alimentación.....	59
Figura 40 - Terminal v1.9b.....	61
Figura 41 - Diagrama de relación entre cinemática inversa y directa.....	64
Figura 42 - Diagrama con los sistemas de referencia, eslabones y ejes del robot.....	65
Figura 43 - Diagrama con los sistemas de referencia y los ejes del robot .....	65
Figura 44 - Esquemático de los 3 primeros GDL del robot.....	69
Figura 45 - Configuraciones codo arriba y abajo.....	70
Figura 46 - ANEXO II. Secuencia de dibujado de los eslabones .....	83
Figura 47 - ANEXO II. Representación del robot completo con la pinza abierta en su extremo.	84
Figura 48 - ANEXO II. Captura de pantalla del simulador.....	84

# Índice de tablas

Tabla 1 - Comparativa de robots industriales educativos comerciales .....	15
Tabla 2 - Tabla longitud prolongadores de servos. ....	27
Tabla 3 - Tabla longitud de los cables de señal de posición.....	28
Tabla 4 - Tabla servomotores, tipos y características.. ....	29
Tabla 5 - Problema con módulo Arduino seleccionado .....	33
Tabla 6 - tabla resumen de entradas analógicas y digitales disponibles y requeridas .....	34
Tabla 7 - correspondencia de pines y servos al conector hembra del brazo.....	56
Tabla 8 - correspondencia de pines y servos al conector macho de 4 pines. ....	56
Tabla 9 - correspondencia de pines y servos al conector macho de 5 pines .....	56
Tabla 10 - Parámetros Denavit-Hartenberg .....	66
Tabla 11 - ANEXO I/A. Tiempo de impresión piezas Eslabón 0 .....	79
Tabla 12 - ANEXO I/A. Tiempo de impresión piezas Eslabón 1 .....	79
Tabla 13 - ANEXO I/A. Tiempo de impresión piezas Eslabón 2 .....	79
Tabla 14 - ANEXO I/A. Tiempo de impresión piezas Eslabón 3 .....	80
Tabla 15 - ANEXO I/A. Tiempo de impresión piezas Eslabón 4 .....	80
Tabla 16 - ANEXO I/A. Tiempo de impresión piezas Eslabón 5 .....	80
Tabla 17 - ANEXO I/A. Tiempo de impresión piezas Pinza .....	80
Tabla 18 - ANEXO I/A. Tiempo de impresión por eslabones y tiempo de impresión de A-Bot completo. ....	81
Tabla 19 - ANEXO I/B. Presupuesto prototipo A-Bot completo.....	81



# Índice de ecuaciones

Ecuación 1. Obtención de la Matriz de transformación D-H.....	66
Ecuación 2. Matriz de transformación D-H .....	66
Ecuación 3. Resultados obtenidos de la Matrices de transformación ( ${}^{i-1}A_i$ ). ....	67
Ecuación 4. Producto para la obtención de la matriz de transformación T.....	67
Ecuación 5. Matriz de Transformación T.....	67
Ecuación 6. Forma de las soluciones de la cinemática inversa. ....	68
Ecuación 7. Forma de las soluciones de posicionamiento. ....	68



# Capítulo 1

## Introducción y objetivos

## Capítulo 1: Introducción y Objetivos

### 1.1. Introducción

Debido a la existencia de una sociedad y un mercado cada vez más exigente y competitivo, la industria se ve obligada a tender cada vez más hacia la automatización, aportando mejoras en diferentes campos:

- En lo referente a la economía, aporta una mejora en la cadena productiva (mejora de procesos de transformación de la materia, disminución de costes de producción, eliminación de altas cantidades de residuos...), a su vez maneja información del mercado (determinación de los mejores proveedores, determinación de los precios y calidades esperadas, oferta, demanda..), y responde eficientemente ante cambios (ajustes internos en la cadena productiva, manejo eficiente de los procesos..).
- En lo referente a la ingeniería, aporta seguridad en las operaciones (evitar daños al hombre, equipos, ambiente, mantiene las operaciones de manera segura y continua...), comodidad de la operación (facilita la operación del hombre, evita labores tediosas que provocan un cansancio del operario disminuyendo la calidad del producto...), y ayuda a conocer mejor el proceso (analizando datos, generando modelos...)

Es por todo ello por lo que la implantación de robots industriales ha crecido considerablemente a lo largo del tiempo en la industria, ya que son una herramienta cada vez más relevante, diseñada para la automatización.

Entre las aplicaciones de un robot industrial se pueden encontrar dos grandes grupos:

- Manipulación (ensamblado y desensamblado, paletizado, empaquetamiento, atención de máquinas,...)
- Procesado (soldadura [Al arco, por puntos...], aplicación de sprays, gel..., mecanizado, corte...)

El siguiente gráfico muestra una orientación de la proporción de robots ya instalados a nivel mundial que se dedican a una determinada tarea:

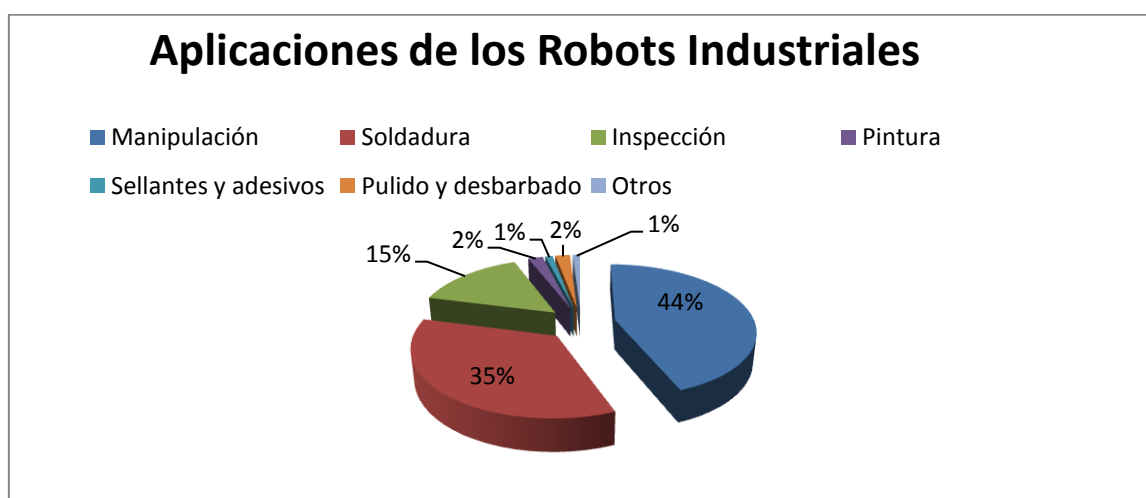


Figura 1 - Gráfico de las aplicaciones de los Robots Industriales.

A la vista de todo lo anterior, no es de extrañar, que cada vez sea más necesario el estudio en Ingeniería de este tipo de robots, desde su morfología, hasta su control y funcionamiento. Es en este punto donde tiene cabida este proyecto final de carrera, cuyo objetivo principal es:

**El diseño, montaje, control y puesta en funcionamiento de un brazo robótico de bajo coste para uso docente, facilitando la adquisición de conocimientos y dotando al estudiante de una mayor motivación al poder probar lo aprendido de una manera práctica antes de pasar al manejo de Robots Industriales reales en posibles sesiones posteriores de prácticas.**

## **1.2. Prototipos de brazos robóticos para docencia, existentes en el mercado y objetivos a lograr.**

Antes de marcar unos objetivos, se debe examinar lo que ya existe en el mercado teniendo en cuenta el tipo de prototipo que se desea diseñar y construir, es por ello por lo que se necesita conocer qué se entiende por robot industrial.

















La Federación Internacional de Robótica (IFR) ISO/TR 83737 Sep.1998 da la siguiente definición:

*“Por robot industrial de manipulación se entiende a una máquina de manipulación automática, re-programable y multi-funcional con tres o más ejes que pueden posicionar y orientar materias, piezas, herramientas o dispositivos especiales para la ejecución de trabajos diversos en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento”*

A raíz de dicha definición se puede obtener una serie de objetivos que deben cumplir los prototipos orientado a docencia:

- “Tres o más ejes”: debe tener mínimo seis grados de libertad (6 G.D.L.) tres para posicionar y tres para orientar.
- “Re-programable”: se debe poder programar una tarea y debe ser capaz de seguir unas trayectorias determinadas, es decir, se necesita que sea programable y servocontrolado, con control en posición para asegurar que realmente realiza la trayectoria determinada obteniendo su posición en todo momento.
- “Multifuncional”: debe ser capaz de agarrar objetos con diferentes formas, y realizar tareas con diferente finalidad (corte, soldadura...).

Una vez claros los objetivos básicos buscados, se expone a continuación la Tabla 1 en la que se realiza una comparativa de las características de los principales brazos robóticos comerciales para educación existentes en el mercado, logrando obtener una idea general de los tres puntos anteriores de una manera gráfica y rápida.

Tipo	G.D.L	Programable	Multifuncional*	Realimentación en Posición	Fabricante	Precio
 <b>Kit brazo robot</b>	5				Super Robótica	52.45 €
					Cebek	51.96 €
					Steren	63.47 €
 <b>AL5A Robotic Arm</b>	5				Super Robótica	413,75 €
					lynxmotion	386.86 €
 <b>Youbot</b>	5				KUKA	21,512.94 €
 <b>A-Bot</b>	6				Antonio Castro Gómez	<b>177.138€</b>

**Tabla 1 - Comparativa de robots industriales educacionales comerciales**

\*Al tratarse para aplicaciones docentes, sólo será necesario centrarse en la manipulación. En tareas como corte, soldadura,... carecen de sentido en esa etapa de aprendizaje (primera toma de contacto con un robot industrial).

Tal y como se observa en la Tabla 1, el prototipo a diseñar ha de cumplir todos los objetivos extraídos de la definición de robot industrial. De todos los modelos mencionados el robot de KUKA es el único que, exceptuando los 6 G.D.L. cumple todo los requisitos, pero tiene un gran inconveniente y es su alto coste a la hora de adquirir uno, por lo tanto, aquí es donde aparece el siguiente objetivo a cumplir para el prototipo a diseñar, ha de ser de bajo coste y de licencia libre, es decir, todo el diseño será realizado con aplicaciones libres y será colgado en internet totalmente documentado para que todo usuario que quiera o necesite un brazo robótico pueda adquirirlo sin necesidad de invertir una alta cantidad de dinero como es el caso de KUKA, se debe tratar de un brazo robótico OpenSource.

Finalmente, un último objetivo sería conseguir el control del brazo mediante lectura y escritura por puerto serie, donde se utilizará un conector USB para dicha comunicación. Esto permite, hoy en día, realizar el control directamente con un ordenador ya que el conector USB lo traen de serie y no, como en el caso de los conectores DB9, que se necesitan convertidores adicionales a la entrada del puerto USB para algunos ordenadores. De este modo, cumpliendo este último objetivo, se eliminan dichos convertidores.

### 1.3. Fases de desarrollo:

En este apartado se expone de forma concisa las diferentes fases por las que se ha pasado para conseguir la elaboración de este proyecto, representando en un gráfico, el tiempo aproximado dedicado a cada una de las partes sobre el tiempo total empleado en su realización.

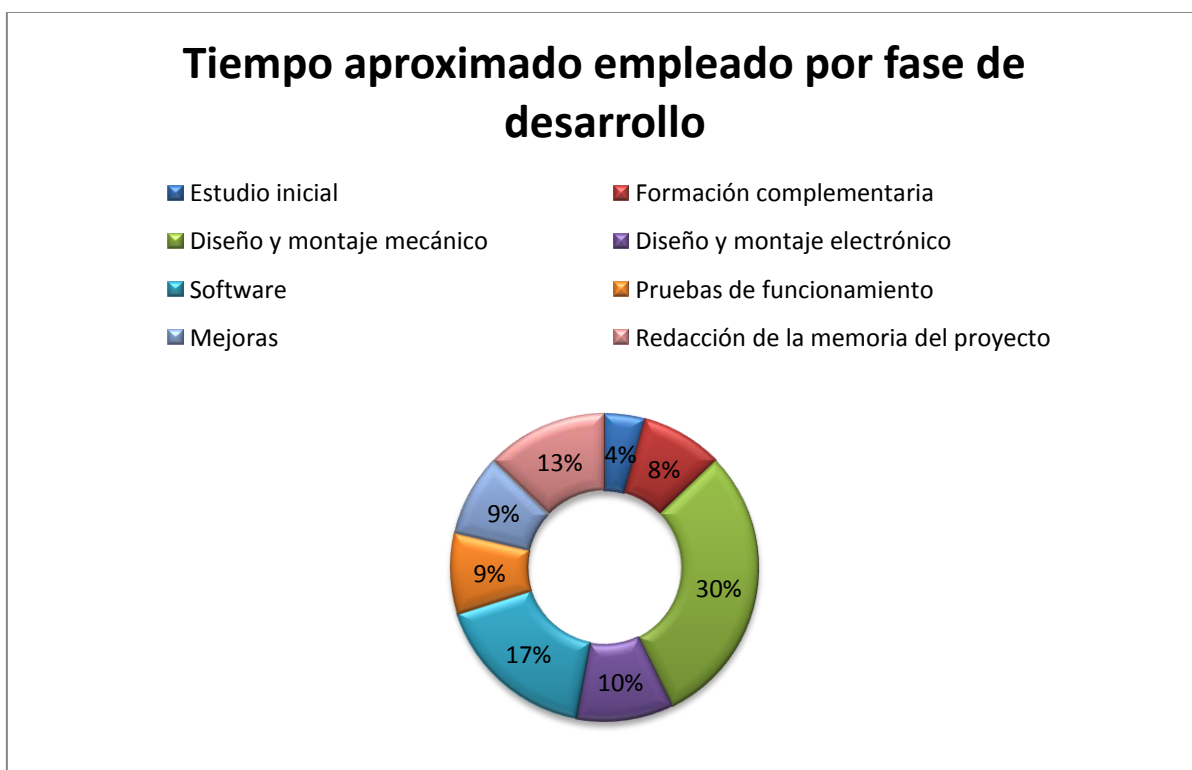


Figura 2 - Gráfico de tiempo aproximado empleado por fase de desarrollo.



**Estudio inicial:** antes de comenzar a realizar el proyecto ha sido necesario realizar un estudio de los tipos de brazos robóticos que existen en el mercado con fines educativos y si sería necesario realizar un prototipo aún más completo y barato que los existentes, decidiendo si continuar con la elaboración del mismo o por el contrario no sería rentable.

**Formación complementaria:** además de la formación adquirida durante la carrera, ha sido necesario ampliarla entrando en el mundo del diseño de piezas por ordenador, adquiriendo destreza en el manejo de programas libres desconocidos hasta el momento, además de conocer cómo funcionan las impresoras 3D OpenSource con las que se realizará el prototipado rápido y económico en ABS de las piezas diseñadas. En cuanto a la parte electrónica, es necesario la obtención de conocimientos de programación en Arduino, lenguaje similar a C (dado en la formación previa) y con el que se creará el control del brazo posteriormente.

**Diseño y montaje mecánico:** Esta fase es una de las principales y una de las que más tiempo va a requerir, puesto que de ella depende la forma, la estética y los posibles problemas que se puedan acarrear en cada movimiento en función del tipo de diseño que se realice para cada eslabón.

Para la realización del diseño es indispensable conocer el tipo de actuadores que se van a emplear (servomotores), pues muchas piezas tienen una determinada forma y dimensión debido al tamaño y hueco que ocupa el servomotor una vez colocado dentro del brazo en su posición final, asegurando su correcta sujeción.

También, en esta etapa, se irán montando las piezas ya diseñadas e impresas con la finalidad de comprobar el correcto funcionamiento de las mismas. En el caso que no lo cumplan se deberá corregir el error cometido, realizando de nuevo diseño, impresión y prueba hasta que funcione, pudiendo en ese caso comenzar con los siguientes diseños.

Antes de comenzar a montar los servomotores, éstos deben ser modificados sacando la señal del potenciómetro que indica la de posición mediante un cable externo.

**Diseño y montaje electrónico:** En esta etapa se realizará la parte hardware del control del brazo robótico, para ello es necesario conocer la placa y el microcontrolador que se va a utilizar a fondo, viendo las prestaciones que nos brinda y las que necesitamos, con el fin de obtener y construir (si fuera necesario), un circuito electrónico complementario conectado a la placa utilizada.

En este caso se ha optado por multiplexar las señales de posición, pues la placa usada posee menos entradas analógicas de las que en realidad se necesitan para leer la posición. Más adelante se expondrá el por qué se optó por esa placa y no otra con más entradas disponibles.

**Software:** Una vez obtenido todo el diseño se desarrolla mediante la programación del microcontrolador seleccionado, el control del brazo por el puerto serie, programando un protocolo de comunicación con el brazo para su posterior control a través de los comandos leídos y escritos creados.

**Pruebas de funcionamiento:** tras la finalización de las etapas de diseño y montaje, es tiempo de realizar todo tipo de pruebas exhaustivas al brazo robótico para comprobar su correcto

funcionamiento en todo momento, creando finalmente una secuencia básica de manipulación de un objeto a modo de prueba.

**Mejoras:** esta parte permite analizar el proyecto en conjunto y realizar sobre el mismo una serie de modificaciones, obteniendo un brazo robótico más robusto y completo que el diseñado inicialmente. Ésta, es una fase con una duración indefinida, dependiente del tiempo del que se disponga para su realización y de la imaginación e ingenio del autor del mismo, puesto que una vez que se tiene un diseño, una mejora lleva a otra y así sucesivamente.

**Redacción de la memoria del proyecto:** con el proyecto finalizado y probado se debe realizar una memoria en la que se documenta todo lo realizado a lo largo del proyecto, indicando problemas, soluciones, mejoras futuras y justificando el por qué se hizo de esa determinada manera.

## 1.4. Medios empleados:

Los medios empleados para la realización del proyecto, son los siguientes:

### SOFTWARE

- OpenScad: programa de licencia libre para el diseño de las piezas del brazo robótico.
- FreeCAD: programa de licencia libre para la creación de los planos de montaje en la memoria.
- Replicatorg: interfaz de comunicación con las impresoras 3D OpenSource, usado para la impresión de las piezas diseñadas.
- Programa Terminal v1.9b. Sirve para comunicar el brazo robótico con el PC, enviando y leyendo datos por el puerto serie.
- Microsoft Word para redactar la memoria.
- OrCAD para el rutado de la placa de electrónica complementaria.
- Ordenador con sistema operativo Ubuntu y Windows 7.

### HARDWARE

- Impresoras Thing of Matic del laboratorio de Sistemas y Automática.
- Placa Arduino UNO con microcontrolador ATMEGA 328P de Atmel.
- Cable USB-serie. Sirve tanto para descargar software a la placa Arduino UNO como para controlar el brazo robótico desde el PC.
- Servos Futaba S3003, Futaba S3305, Tower Pro MG90S, Hitec HS-55 (todos con sus gomas, platos y tornillos de sujeción correspondientes).
- 7 tornillos y 7 tuercas de métrica M3 20mm de longitud
- 32 tornillos de estrella 5x20mm para madera. Se usará para unir las piezas fijas impresas en ABS.
- 22 tornillos de métrica M2x6mm para montar los platos de los servos en las piezas impresas en ABS.
- Bridas blancas y negras para sujetar los cables a la estructura.
- Mangueras recoge cables para tener flexibilidad en las articulaciones del brazo robótico y tener ordenado los cables que van a la placa.

- Cable triple plano para servos. Utilizado para crear prolongadores que separa alimentación de la señal PWM.
- Cable unipolar para sacar la señal de posición (vale un hilo del cable de tres hilos anterior).
- Cable bipolar de  $0.7\text{mm}^2$  para alimentación conjunta de los servos.
- Tubo termo retráctil de 6 y de 2 mm de diámetro.
- Multiplexor 74HC4052 con zócalo de 16 pines para multiplexar las señales de posición.
- 111 pines macho y 32 pines hembra.
- Dos clemas para PCB.
- Placa fotosensible de fibra de vidrio para generar circuito de electrónica complementaria.
- Fuente de PC con toma de 5V para alimentar los servomotores (y la placa en ausencia de PC).
- Pegamento SuperGlue3 (LOCTITE).
- Cinta aislante.
- Estaño y soldador de 11W con soporte y esponja.



# Capítulo 2

## Diseño mecánico

## Capítulo 2: Diseño mecánico

### 2.1. Componentes imprimibles

Todas las piezas imprimibles que constituyen cada uno de los eslabones, han sido pensadas y diseñadas conforme a tres objetivos:

- Tener en cada eslabón la mayor rigidez posible en la unión de todas las piezas que lo componen.
- Tener el mayor ahorro de plástico posible, evitando realizar eslabones muy pesados favoreciendo el movimiento del mismo y disminuyendo el tiempo de impresión de las piezas.
- Tratar que tras el montaje completo del brazo, el resultado sea lo más estético posible, dándole una forma bonita a la vez que cumple con los objetivos marcados.

#### 2.1.1. Eslabones y movimiento relativo entre ellos

A continuación, se da una breve justificación de la elección de la forma correspondiente a cada eslabón que compone el brazo robótico indicando su movimiento relativo con el siguiente y sus principales características.

##### 2.1.1.1. Eslabón 0

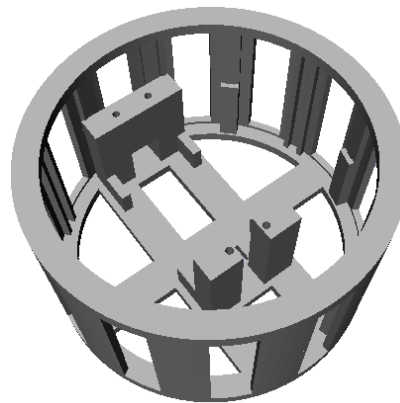


Figura 3 - Eslabón 0 montado sin servos (Base del brazo robótico)

- **Movimiento relativo eslabón 0-1:** Rotacional ( $0^\circ$  a  $360^\circ$ )
- **Número de piezas impresas que lo componen:** 3

Como se puede observar se trata de la base del robot sobre la cual se va a apoyar el resto del brazo. Dicho eslabón se debe asegurar a una superficie fija, de manera, que el movimiento del brazo no lo vuelque.

Las paredes laterales del eslabón poseen una serie de agujeros rectangulares, realizados para poder observar desde fuera el movimiento del servomotor que mueve el eslabón uno y la correcta colocación de los cables una vez instalados los servomotores. Dichas paredes poseen un

refuerzo a lo largo de ellas además de poseer un encaje macho-hembra, dotando la estructura de una mayor rigidez gracias al encaje.

Se ha optado por seleccionar una forma cilíndrica, debido a que, el movimiento relativo del eslabón cero con el eslabón uno es de rotación, quedando más estético y permitiéndonos conocer de forma intuitiva el movimiento que tendrá el siguiente eslabón.

El servomotor sujeto en este eslabón permite girar al eslabón uno  $180^\circ$ , mientras que el servomotor enfrentado y sujeto en el eslabón uno, permite girar el resto (de  $181$  a  $360^\circ$ ).

### 2.1.1.2. Eslabón 1

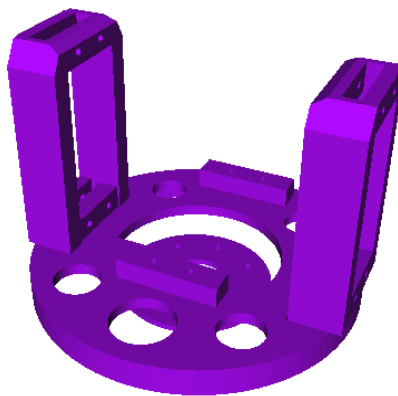


Figura 4 - Eslabón 1 montado sin servos

- **Movimiento relativo eslabón 1-2:** Rotacional ( $0^\circ$  a  $180^\circ$ ).
- **Número de piezas impresas que lo componen:** 9

Éste eslabón se caracteriza por la forma en la que se colocan sobre él los servos. Gracias a dicha colocación obtiene una alta rigidez durante el movimiento. En él, va sujeto el servomotor enfrentado al colocado en el eslabón cero (que permite un giro de  $181$  a  $360^\circ$ ) y el que permite el giro del eslabón dos (colocados de forma vertical para tener un mayor margen de movimiento y una mejor sujeción del siguiente eslabón al servo). El eslabón se ha diseñado de tal manera que para el movimiento del eslabón dos se podrían añadir dos servos en lugar de uno aportando mayor par. En el prototipo de brazo propuesto en este proyecto solo se utilizara un servomotor para dicho eslabón.

Los agujeros apreciados en la base serán utilizados para la colocación de los cables.

La arandela agujereada que se observa en la base del eslabón, es colocada entre los dos servos que crean el movimiento relativo y permiten el giro de  $360^\circ$ , permitiendo sujetar ambos servos atornillándolos entre sí de forma recta.

### 2.1.1.3. Eslabón 2

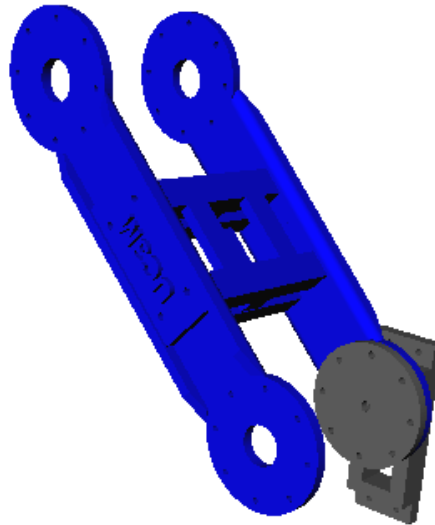


Figura 5 - Eslabón 2 montado sin servos

- **Movimiento relativo eslabón 2-3:** Rotacional ( $0^\circ$  a  $180^\circ$ )
- **Número de piezas impresas que lo componen:** 10

En este eslabón se busca que tienda a curvarse lo menos posible, pues es uno de los que dan al brazo alcanzabilidad. Para ello se ha añadido a cada lado de las piezas que forman los laterales un refuerzo a lo largo del eslabón que lo impide.

Con la finalidad de evitar descoordinación entre los laterales (que en el movimiento uno gire más que el otro) y para darle una mayor rigidez, se ha introducido en la zona central un módulo agujereado y hueco por dentro. Este módulo será uno de los esenciales en la colocación de los cables, pues se aprovechará su estructura y su hueco interior para alojarlos.

Al estar los laterales compuestos por dos piezas cada uno, se ha añadido una pieza rectangular (en la que por estética se le ha puesto UC3M), que evita que las piezas que componen el lateral se separen sujetando, además, el módulo central uniendo todo con tornillos, haciéndolo aún más compacto.

Al mover el eslabón con un servo únicamente, para sujetar el lado que no tiene servo, se ha diseñado un módulo (módulo gris de la figura 5) que simula la parte frontal de un servo y hace que el eslabón gire por ambos lados en un mismo eje, el eje del servomotor.

#### 2.1.1.4. Eslabón 3

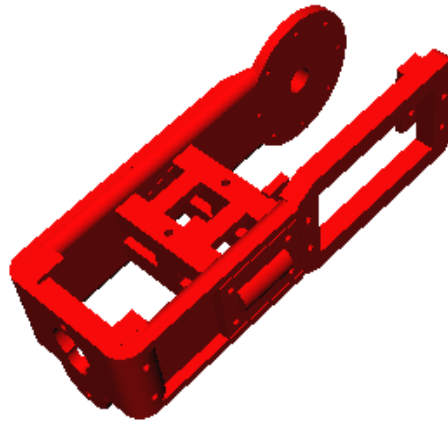


Figura 6 - Eslabón 3 montado sin servos

- **Movimiento relativo eslabón 3-4:** Rotacional ( $0^\circ$  a  $180^\circ$ ).
- **Número de piezas impresas que lo componen:** 11

Las características que se buscan en este eslabón, son las mismas que las buscadas en el eslabón dos, es por ello por lo que la forma es similar, incluyendo las piezas laterales y el módulo central agujereado y hueco.

La principal diferencia con el eslabón anterior es que uno de sus extremos va conectado a lo que se denomina “muñeca” del robot, por ello termina en una pieza plana especialmente diseñada para sujetar en ella el microservo correspondiente al movimiento del eslabón 4.

El extremo contrario al de la “muñeca” y diferente al del eslabón 2, difiere debido a que en esa zona se aloja el servo que da el movimiento al eslabón.

#### 2.1.1.5. Eslabón 4

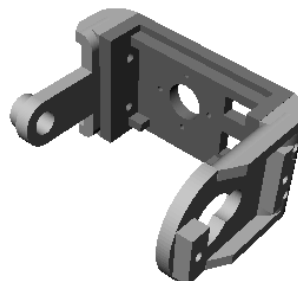


Figura 7 - Eslabón 4 montado sin servos

- **Movimiento relativo eslabón 4-5:** Rotacional ( $0^\circ$  a  $180^\circ$ ).
- **Número de piezas impresas que lo componen:** 3



El diseño de los siguientes eslabones (4 y 5) se ha realizado de modo que queden unidos de forma compacta, ocupando el menor espacio posible, pues son los correspondientes a la “muñeca del robot”.

Otro de los requisitos de diseño es que los tres servos correspondientes a los tres grados de libertad de la muñeca, se coloquen en los eslabones de manera que se pueda aplicar el método del desacoplo cinemático, simplificando de este modo el cálculo de la cinemática inversa, obteniendo con los tres primeros grados de libertad, la posición, y con los tres siguientes (muñeca) la orientación de forma sencilla.

Atendiendo a éstos requisitos, la forma final del eslabón cuatro es la que se observa en la figura 7, donde en uno de sus extremos se sujeta el servo correspondiente al movimiento del siguiente eslabón, mientras que en el otro se ha realizado una pieza que permita sujetar ambos eslabones girando siempre por el eje de giro del servomotor.

El servomotor encargado del giro del eslabón se encuentra situado en el eslabón anterior (eslabón 3) como se indicó anteriormente.

#### 2.1.1.6. Eslabón 5 y 6

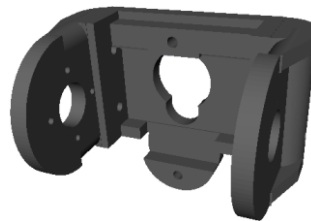


Figura 8 - Eslabón 5 montado sin servos

- **Movimiento relativo eslabón 5-6:** Rotacional ( $0^\circ$  a  $180^\circ$ ).
- **Número de piezas impresas que lo componen:** 3

Las pautas de diseño seguidas son las mismas que para el eslabón 4. La diferencia reside en que se trata del último eslabón diseñado y conectado al brazo robótico, ya que la pinza no es considerada como un eslabón.

El eslabón 6 del brazo, que es el que conecta directamente con la pinza, no se ha diseñado, porque se ha considerado como eslabón el propio eje del servomotor sujeto en el eslabón cinco (pieza frontal del eslabón cinco diseñada para dicho fin) y correspondiente al movimiento del eslabón seis (aspas del eje).

### 2.1.1.7. Pinza

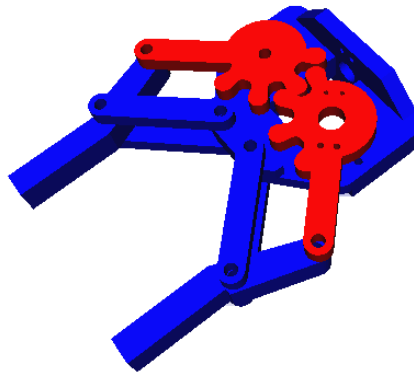


Figura 9 - Pinza montada sin servo

La pinza es la única parte del brazo robótico, cuyo diseño, ha sido descargado de internet, concretamente de la web [www.thingiverse.com](http://www.thingiverse.com), dicha web corresponde a la comunidad en la que los usuarios suben sus diseños de manera libre (Open Source), para que cualquiera que tenga una impresora 3D pueda hacer uso de ellos. El brazo robótico diseñado a lo largo de este proyecto ha sido subido a la misma.

El enlace de la pinza es el siguiente:

<http://www.thingiverse.com/thing:7109>

## 2.2. Componentes no imprimibles

En este apartado, se expone de forma detallada, todo el material complementario no imprimible necesario para montar el brazo, dejando todo listo para poder abordar las instrucciones de montaje que se encuentran en el capítulo 4.

### 2.2.1. Material de soporte complementario

#### - *Cables de señal PWM y alimentación de los servos:*

Todos los servomotores utilizados traen de fábrica un cable de 30cm de longitud con un conector hembra de tres pines conectado en su extremo. Debido a que los cables han de ir sujetos a la estructura del brazo evitando que puedan enredarse o cortarse durante el movimiento del mismo, es necesario añadir un prolongador a cada servo con una longitud determinada que nos permita ir desde el servomotor hasta la placa donde irán conectados los cables. Dicho prolongador se va a realizar con un cable triple plano (como el que traen los servos, igual sección también), en cuyo extremo se colocará un conector de tres pines macho permitiéndonos conectarlo al conector hembra que traen de serie, recubriendo las soldaduras con tubo termoretráctil para evitar cortocircuitos. En el otro extremo, por el momento, no se pondrá ningún conector, pues habrá que realizar una serie de modificaciones en el mismo durante la etapa de montaje como se verá en el capítulo 4. (Véase figura 10)



Figura 10 - Prolongador de servos

En la tabla 2 se indican las dimensiones de los cables correspondientes para cada servo. Se ha adjudicado a cada servo una letra, comenzando desde la base hasta la pinza de forma progresiva, de modo que el servo A es el anclado en el eslabón cero (A1 sujeto en eslabón 0 y A2 sujeto en eslabón 1) y el servo G corresponde a la pinza.

Servo	Longitud Prolongador (m)*
A1	0.70
A2	0.70
B	0.70
C	0.70
D	0.85
E	1.05
F	0.85
G	1.30

Tabla 2 - Tabla longitud prolongadores de servos.

\*Aunque se explique de forma detallada como sujetar el cable, siempre varía algo la posición, haciendo variar la longitud del cable final, por ello las longitudes expresadas son longitudes aproximadas y no exactas.

En lo que respecta a los cables de alimentación, es necesario la obtención de 1m de cable bipolar  $0.7\text{mm}^2$  flexible. Todas las alimentaciones de los servos serán soldadas al mismo cable dejando sólo dos hilos para alimentación y protegiendo las soldaduras con tubo termoretráctil. Esto nos simplificará la colocación del cableado considerablemente, además de reducir la manguera de cable que se obtiene al final y que va hacia la placa de arduino y la placa de electrónica complementaria.

- **Cable de señal de posición:**

Al igual que ocurre con los cables de señal PWM, las medidas que se indican en la tabla 3 son aproximadas. A estos cables unipolares (de igual sección que los que lleva el servo de serie) no hay que soldarles ningún tipo de conector, pues un extremo va soldado directamente al servomotor y el otro hay que manipularlo durante el montaje como se expone en el capítulo 4.

Servo	Longitud cable señal posición (m)
A1	0.75
A2	0.75
B	0.75
C	0.75
D	0.90
E	1.10
F	0.90
G	1.35

Tabla 3 - Tabla longitud de los cables de señal de posición.

Todos los servos se sujetan a los eslabones con los tornillos que vienen de serie en el servomotor para tal fin. El resto de elementos utilizados para el montaje se pueden observar en la figura 11.

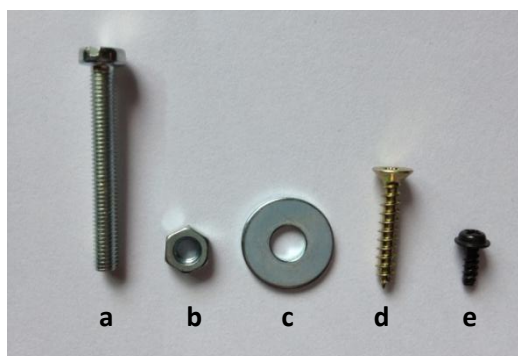


Figura 11 - Tornillería con tuercas y arandelas

- |  |   |
|--|---|
| a) Tornillo de métrica M3x25mm                         | -> Uso en pinza y eslabón 2.                                  |
| b) Tuerca de métrica M3                                | -> Uso en pinza y eslabón 2.                                  |
| c) Arandela: $\Phi$ exterior=12mm, $\Phi$ interior=5mm | -> Uso en eslabón 2.  |
| d) Tornillo helicoidal 5x15mm (rosca 2,5mm)            | -> Uso en uniones de piezas impresas                          |
| e) Tornillo de métrica M2x6mm                          | -> Uso en uniones entre piezas impresas y ejes de los servos. |

**Nota:** durante el montaje hay partes en las que es necesario cortar los tornillos porque sobresalen demasiado, todo aparece indicado en el capítulo 4.

### 2.2.2. Servomotores modificados

Tras buscar y comparar servos, se ha decidido utilizar cuatro tipos para la composición del brazo. A continuación se indica el por qué se ha optado por esos servomotores y se adjunta la tabla 4 donde se pueden comprobar las principales características de los mismos.

- Futaba S3003: la elección de este servomotor es debido a su calidad precio, se trata de un servo de dimensiones estándar y es barato en comparación con otros de par similar.

- Futaba S3305: este servo es el encargado de mover todo el brazo a partir del eslabón 2 (incluido). Con anterioridad se probaron en su lugar otros servos y finalmente se decidió poner éste (aunque es el más caro de todos los utilizados) por su alto par que proporciona, y robustez que aporta. Moviendo el brazo en la peor configuración (totalmente extendido) sin problemas.
- Tower Pro MG90S y Hitec HS-55: se trata de dos microservos, de par elevado para el tamaño que tienen, estos servos van situados en la muñeca y en la pinza del brazo, al ser más pequeños, pesan menos, ofreciendo menos resistencia al movimiento del brazo y dan la fuerza suficiente para mover el eslabón al que está conectado.





Servo	Peso (g)	Velocidad (seg/60°)	Torque (kg·cm)	Tamaño (mm)	Engranajes
 <b>Futaba S3003</b>	37.2	0.23	4,8V -> 3.2 6 V -> 4.1	40x20x38.1	Plástico
 <b>Futaba S3305</b>	46.5	0.2	4,8V -> 8 6 V -> 8.9	40.4x19.6x36	Metálicos
 <b>Tower Pro MG90S</b>	13.4	0.1/0.08	4,8V -> 1.8 6 V -> 2.2	22.8x12.2x28.5	Metálicos
 <b>Hitec HS-55</b>	8	0.17/0.14	4,8V -> 1.1 6 V -> 1.3	22.8x11.6x24	Plástico

Tabla 4 - Tabla servomotores, tipos y características..

Uno de los principales objetivos del brazo robótico, es el de leer la posición en la que se encuentra el servomotor. Para llevar a cabo el cumplimiento de dicho objetivo, es necesario modificar los servomotores extrayendo la señal de posición del interior del servo.

El servomotor controla internamente su posición a través de un potenciómetro que gira solidario al eje, lee la posición (pata intermedia del potenciómetro), la compara con el valor de la señal PWM que está recibiendo y actúa en consecuencia, girando el eje hacia la izquierda, hacia la derecha o dejándolo quieto porque ya se encuentre en la posición.

Una vez conocido el funcionamiento del servo, tenemos que abrirlo y soldar el cable unipolar (preparado en el punto anterior) a la patilla central del potenciómetro (señal de posición). Se puede soldar directamente a la patilla ó por el contrario, se busca el punto en la placa interior que tiene esa señal y evitamos desmontar el servomotor entero.

En la siguiente dirección existe un manual que explica paso a paso como se obtendría dicha señal en un servo Futaba S3003.

[http://www.learobotics.com/wiki/index.php?title=Hacked\\_servo](http://www.learobotics.com/wiki/index.php?title=Hacked_servo)

Para el resto de servos el proceso es el mismo. Hay que tener especial cuidado en los microservos (Tower Pro y Hitec), pues para acceder a la señal de la placa hay que sacarla (en los servos Futaba al quitar la tapa, la señal queda a las vistas directamente) y desliar con cuidado los cables que van directos al potenciómetro y al motor. Si no se realiza con extremo cuidado se pueden cortar los cables o aún peor, por un tirón al igual que por sobrecalentamiento al soldar el cable, se puede arrancar una pista de la placa y dejar el servo inservible (véase en figura 12 la diferencia entre la placa del microservo (MG90S) y el servomotor estándar (S3003), donde además se indica la ubicación de la señal de posición (cable blanco en MG90S y soldadura central de las tres paralelas a las soldaduras de los cables que trae el servo S3003 de serie)).

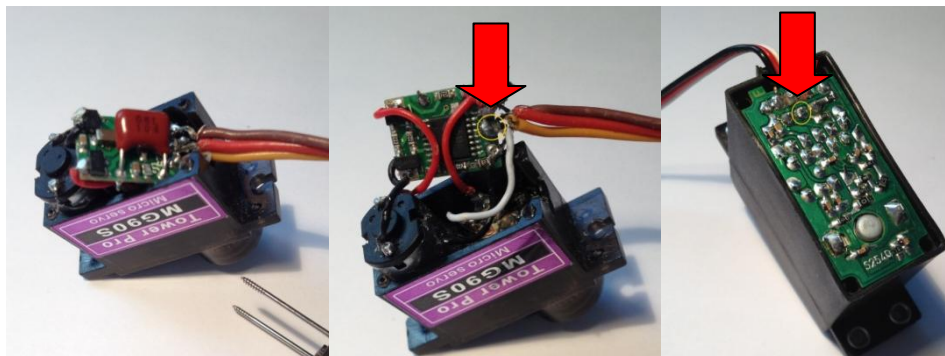


Figura 12 - Diferencias entre la placa de un microservo y un servo estándar

Los pasos a seguir para realizar la modificación básicamente son:

- Abrir el servomotor quitando los cuatro tornillos de la base y estirando de la tapa.
- Localizar la señal del potenciómetro en la placa.
- Pelar el cable y soldarlo con cuidado de no sobrecalentar mucho la zona.
- Realizar una hendidura en la tapa trasera en la zona que salen los cables antes de ponerla y apretar los cuatro tornillos para que cierre bien y se pueda sacar el cable nuevo sin problemas.
- Poner la tapa y apretar los cuatro tornillos.
- Finalmente el servo quedará con 4 cables (tres que trae de serie: señal PWM, VCC y GND más el nuevo cable de posición).



# Capítulo 3

## Diseño electrónico y control



## Capítulo 3: Diseño electrónico y control

### 3.1. Arduino UNO

Para la realización de la comunicación y control del brazo robótico, se ha optado por la utilización de la plataforma Arduino al tratarse de una plataforma de hardware libre como el brazo robótico. Se basa principalmente en una placa con un microcontrolador y un entorno de desarrollo, que ha sido diseñado de forma que no se necesita tener grandes conocimientos de electrónica para su uso, pues su principal objetivo es facilitar el uso de la electrónica en proyectos multidisciplinarios.

El microcontrolador escogido para la realización de este prototipo es Arduino UNO R3 con el microprocesador ATMEGA 328 de Atmel. La elección de este módulo se debe principalmente a su buena relación calidad-precio (véase el módulo en figura 13).



Figura 13 - Arduino UNO R3

Características principales del módulo:

- Tensión de funcionamiento 5V.
- Voltaje de entrada (recomendada) 7-12V.
- Voltaje de entrada (límites) 6-20V.
- Entradas/salidas digitales 14.
- Entradas analógicas 6.
- Memoria Flash 32KB (ATMEGA328).
- SRAM 2KB (ATMEGA328).
- EPROM 1KB (ATMEGA328).
- Velocidad de reloj 16MHz.
- Conexión con PC mediante USB.

Una de las características por las que también se ha optado por el uso de la plataforma Arduino, es debido a que tiene la capacidad de generar y gestionar automáticamente sin complicar la programación una señal PWM en sus pines, a través de la cual podremos gobernar los servos del brazo sin demasiada complicación.



### 3.2. Electrónica complementaria (componentes característicos)

Una vez que se ha seleccionado el módulo de Arduino a utilizar, hay que diseñar la forma en que se va a conectar la placa con todos los servos y señales de posición que deben ser gobernadas por el módulo con su correspondiente microcontrolador.

El brazo se compone de un total de 8 o 9 servos según el número de servos colocados en el movimiento del eslabón 2 (el brazo robótico implementado es de 8 servos por lo que a partir de ahora sólo se indicarán los 9 servos a modo de recordatorio en tablas, no en explicaciones), cuya señal PWM se generará desde un pin digital y una señal de posición por cada servo, es decir, hay 8 señales de posición que deben ser leídas mediante una entrada analógica del módulo Arduino Uno. Es aquí donde aparece el primer problema al usar este modelo, pues el módulo seleccionado posee 14 entradas/salidas digitales (al utiliza el puerto serie por ser uno de los requisitos del proyecto, el cual corresponde los pines digitales 0 y 1, se tendrían 12 y no 14 I/O digitales) de las cuales usaremos 8 para generar la señal PWM que gobernará a los servos. Este punto no genera conflicto, en cambio el siguiente que se expone si, debido a que el módulo posee 6 entradas analógicas y se requieren 8, una para cada señal de posición (ver tabla 5).

	Arduino UNO (disponibles)	Brazo Robótico (necesarias)
<b>I/O Digitales</b>	12	8 (o 9)
<b>I Analógicas</b>	6	8 (o 9)

Tabla 5 - Problema con módulo Arduino seleccionado

Existen varias soluciones al respecto que se analizan a continuación:

- Solución 1: Cambiar de módulo y seleccionar uno con un mayor número de entradas analógicas. Esta solución es la más sencilla pero se descarta por un gran inconveniente, la diferencia de precio entre un módulo y otro. Como se expuso en el capítulo uno, uno de los principales objetivos a conseguir es realizar un brazo robótico de bajo coste y al realizar el cambio de módulo estaríamos incrementando el precio de forma considerable.
- Solución 2: multiplexación de las señales de posición. Esta solución es más complicada que la solución uno al requerir electrónica complementaria para su realización, pero es una solución más económica y se obtiene el mismo resultado que con la anterior. Es por ello por lo que se aplica.

Para llevar a cabo la solución seleccionada, es necesario añadir un multiplexor a una de las salidas analógicas del módulo Arduino UNO, por lo tanto, hay que diseñar una placa que albergue en ella al multiplexor y permita la conexión directa con la placa de Arduino.

Puesto que el brazo se puede componer por 8 o 9 servos, la placa será diseñada para 9 servos aunque posteriormente se conecten sólo 8.

El siguiente paso a resolver es el tipo de multiplexor que se pondrá, para ello se analiza los pros y los contras de cada uno de ellos:

- 74HC4051 (multiplexor 8/1 (8 entradas, una salida))  
Este multiplexor es capaz de manejar 8 señales de entrada utilizando sólo una entrada analógica de Arduino, la novena señal de posición debe ser manejada por otra entrada, es decir en total se utilizarían dos entradas analógicas de la placa.  
Para indicar al multiplexor que entrada debe salir por su salida, al ser 8 señales requiere de tres bits para su direccionamiento, luego se requieren 3 entradas digitales de la placa de Arduino.
- 74HC4052 (doble multiplexor 4/1 (4 entradas, una salida))  
Este chip contiene en su interior dos multiplexores de 4 entradas y una salida, en total puede manejar 8 señales de entrada, y utilizaría dos entradas analógicas de la placa de Arduino, a cada entrada de la placa le corresponden 4 señales. La novena señal de posición debe ser manejada por otra entrada analógica, por lo que en total se utilizarían 3 entradas analógicas para la lectura de la posición.  
En cuando a la selección de la señal que debe aparecer a la salida del multiplexor, al tratarse de dos multiplexores de 4 entradas (4 señales) se requieren solamente dos bits para su direccionamiento, por lo que, en total, se necesitan dos entradas digitales de la placa Arduino UNO.

En la tabla 6 se puede observar a modo de resumen las entradas analógicas y digitales que necesitamos para cada chip y las que disponemos, por lo tanto, se observa que si se utiliza el chip 74HC4051 se agotan todas las I/O digitales para el caso de 9 servos (una libre para el caso de 8 servos) y solo quedarían libres 4 salidas analógicas. Mientras que si se utiliza el chip 74HC4052 se deja una I/O digital libre para el caso de 9 servos (dos para el caso de 8 servos) a consta de ocupar una entrada analógica más. A la vista de todo esto, se escogerá el chip 74HC4052 ya que para mejoras futuras deja libre una I/O digital si se utilizan 9 servos, pudiéndola usar para sensorizar el brazo o adaptarlo a su entorno de trabajo.

	Arduino UNO* (disponible)	CHIP 74HC4051 (requeridas)	CHIP 74HC4052 (requeridas)
<b>I/O Digitales</b>	3 (brazo 9 servos) 4 (brazo 8 servos)	3	2
<b>I Analógicas</b>	6	2 (brazo 9 servos) 1 (brazo 8 servos)	3 (brazo 9 servos) 2 (brazo 8 servos)

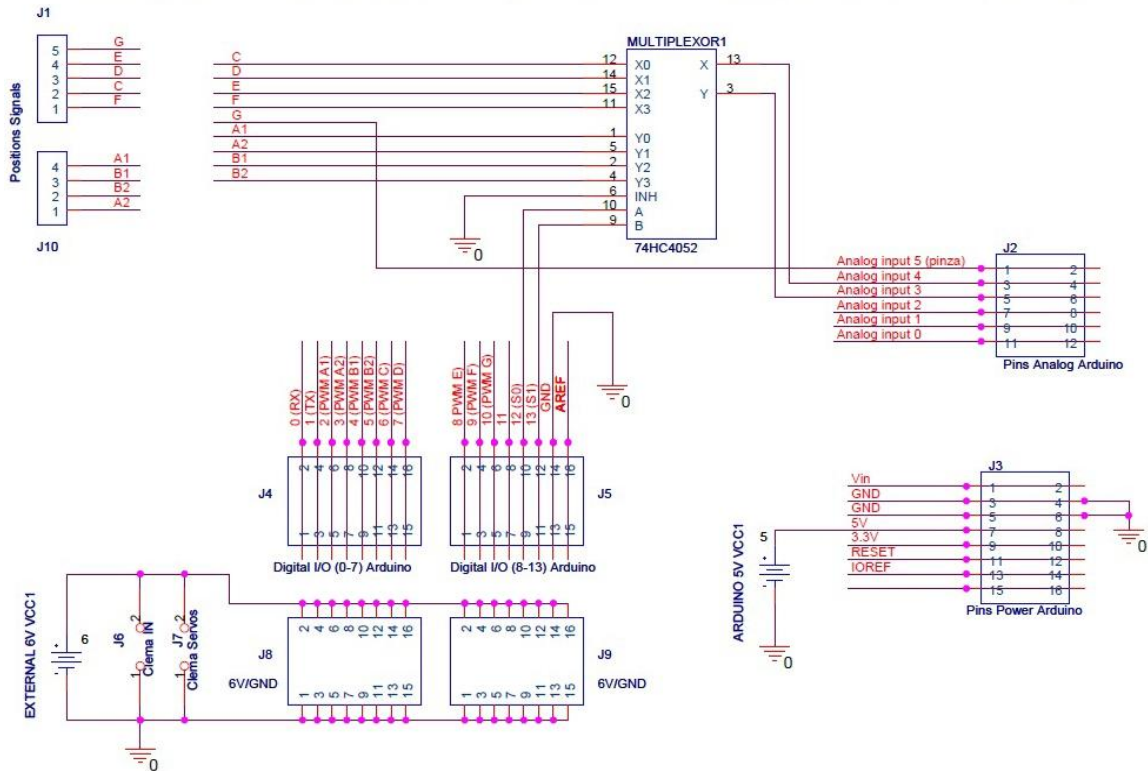
Tabla 6 - tabla resumen de entradas analógicas y digitales disponibles y requeridas

- \* Total 14 digitales -> 8 servos: 14 totales – 8 PWM – 2 puerto serie= 4 digitales libres  
-> 9 servos: 14 totales – 9 PWM – 2 puerto serie= 3 digitales libres

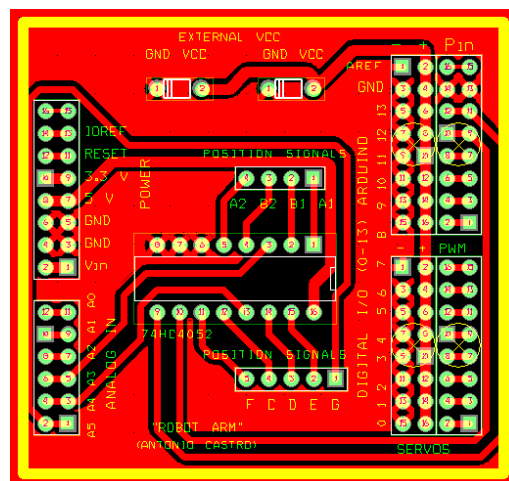
Para simplificar el montaje y conexionado de todas las señales y servos del brazo, se ha diseñado una placa que va conectada directamente a los conectores hembra de Arduino UNO poseyendo todas sus señales, además de nuevas conexiones requeridas para la gobernación del brazo, incluyendo la alimentación.

A continuación se puede observar el diseño de la placa realizado en Orcad Capture CIS (figura 14), Orcad Layout (figura 15), la placa ya implementada con todos los componentes soldados (figura 16), ambas placas conectadas (figura 17) y un resumen de las principales características y mejoras de la misma.

NOTA: los pines 7,8 y 16 del multiplexor, no presentes en el esquemático, se conectan de la siguiente forma: pin 7 y 8 a GND y pin 16 a 5V VCC



**Figura 14 - Esquemático de la placa para electrónica complementaria en Orcad**



**Figura 15 - Diseño de placa para electrónica complementaria en Orcad Layout**

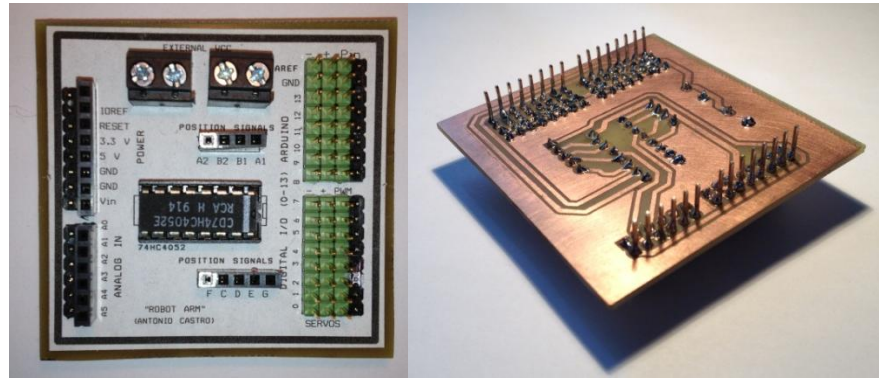


Figura 16 - Placa diseñada implementada y completa con componentes soldados

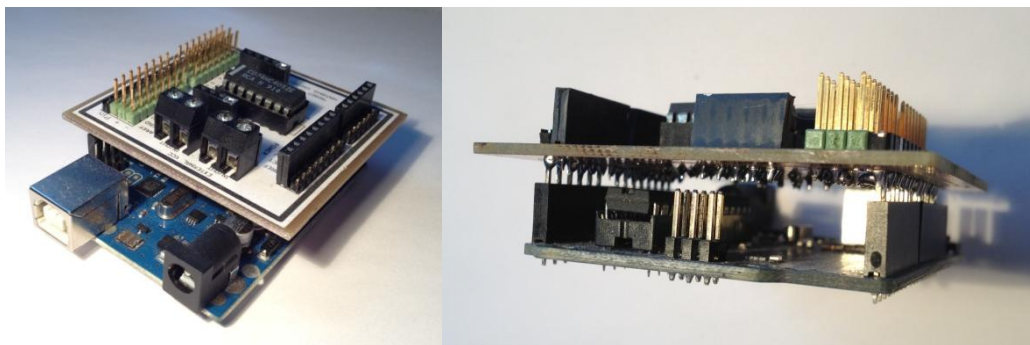


Figura 17 - Placas conectadas listas para el cableado

A continuación se muestran las principales características de la placa diseñada:

- Va conectada directamente a la placa Arduino UNO
- Contiene todos los pines de Arduino UNO accesibles además de contener los pines requeridos para la lectura de posición.
- Cada pin digital se compone de tres conectores macho, permitiendo la conexión del conector hembra que llevan los servos de serie directamente. Los dos pines adicionales al pin digital de Arduino corresponden a la alimentación introducida por las clemas de “EXTERNAL VCC”
- Las clemas “EXTERNAL VCC” son las encargadas de la alimentación de los servos del brazo. En una clema se conecta la fuente y en la otra se conecta la alimentación del brazo.
- Contiene 9 conectores hembra para conectar 9 señales analógicas (la placa está preparada para leer la posición de 9 servos en caso de poner en el eslabón dos, dos servos en vez de uno). Se ha adjudicado a cada servo una letra, comenzando desde la base hasta la pinza de forma progresiva, de modo que el servo A es el anclado en el eslabón cero (A1 sujeto en eslabón 0 y A2 sujeto en eslabón 1) y el servo G corresponde a la pinza.
- Zócalo para conexión del multiplexor 74HC4052.

- Los pines analógicos (A0, A1, A2) y el pin digital 11 quedan libres permitiendo en trabajos futuros la sensorización del brazo o adaptarlo a su ubicación conectando a dichos pines sensores del entorno de trabajo donde esté situado.

### 3.3. Programación y funcionamiento (señales de control)

La programación del microcontrolador ATMEGA 328 de ATMEL insertado en la placa de Arduino UNO, debe realizarse de acuerdo al cumplimiento de los objetivos marcados en el capítulo uno, donde se indica que el brazo robótico debe ser controlado mediante el puerto serie USB, es por ello por lo que se ha creado un programa que lea y envíe información del puerto serie, obedeciendo a las instrucciones en forma de comandos que le llegan creadas para su control. El funcionamiento del programa se puede observar en el flujograma adjunto correspondiente (véase figura:20)

El programa se basa fundamentalmente en tres partes:

- **Movimiento:** esta parte es la encargada de procesar la instrucción de movimiento, llevando al servo definido al ángulo indicado en la instrucción enviada por el puerto serie. Ésta parte no devuelve nada por el puerto serie, únicamente se dedica a mover el servo a una determinada posición.
- **Posición:** este punto es la que hace al prototipo diferenciarse del resto de brazos robóticos existentes en el mercado. Su función es la de leer la posición del servo que se le indique en la instrucción, siendo devuelta por el puerto serie indicando el ángulo en el que se encuentra y el servo al que pertenece dicho ángulo, permitiendo el control en posición en lazo cerrado.

Para dicha función se han implementado dos tipos de lecturas debido a que al observar la señal de posición a través del osciloscopio, se ha comprobado como el servo no filtra la señal PWM totalmente y genera una componente residual que hace que la continua no sea completamente estable. Esto es un problema de cara a la lectura, pues cada vez que el convertidor AD del microcontrolador lea una misma posición, se obtendrán diferentes valores debido a esa componente. Para la solución a este problema se han implementado dos tipos de lectura:

- a) Lectura estimada con un error de tres grados (véase figura: 18).
  - b) Lectura precisa a partir de la estimada con un error de tres grados (véase figura:19).
- **Programación por guiado:** Esta función desconecta los servos, permitiendo manipular el brazo manualmente, mientras se recogen las posiciones con las configuraciones del brazo deseadas, permitiendo una mayor rapidez en la programación, tanto de trayectorias como de puntos.  
Es aquí donde entran en juego los dos tipos de lecturas, pues si la programación por guiado esta activa, es decir, se ha desconectado la señal PWM del servo para manipularlo manualmente, sólo se puede utilizar la lectura estimada, pues no utiliza en su ejecución la señal PWM. Mientras que si la programación por guiado esta desactivada, se utiliza la lectura precisa, pues el brazo estaría controlado por los servomotores y no manualmente, permitiendo comparar la señal PWM con la posición actual estimada leída de cada servo.

Lectura estimada con un error de tres grados

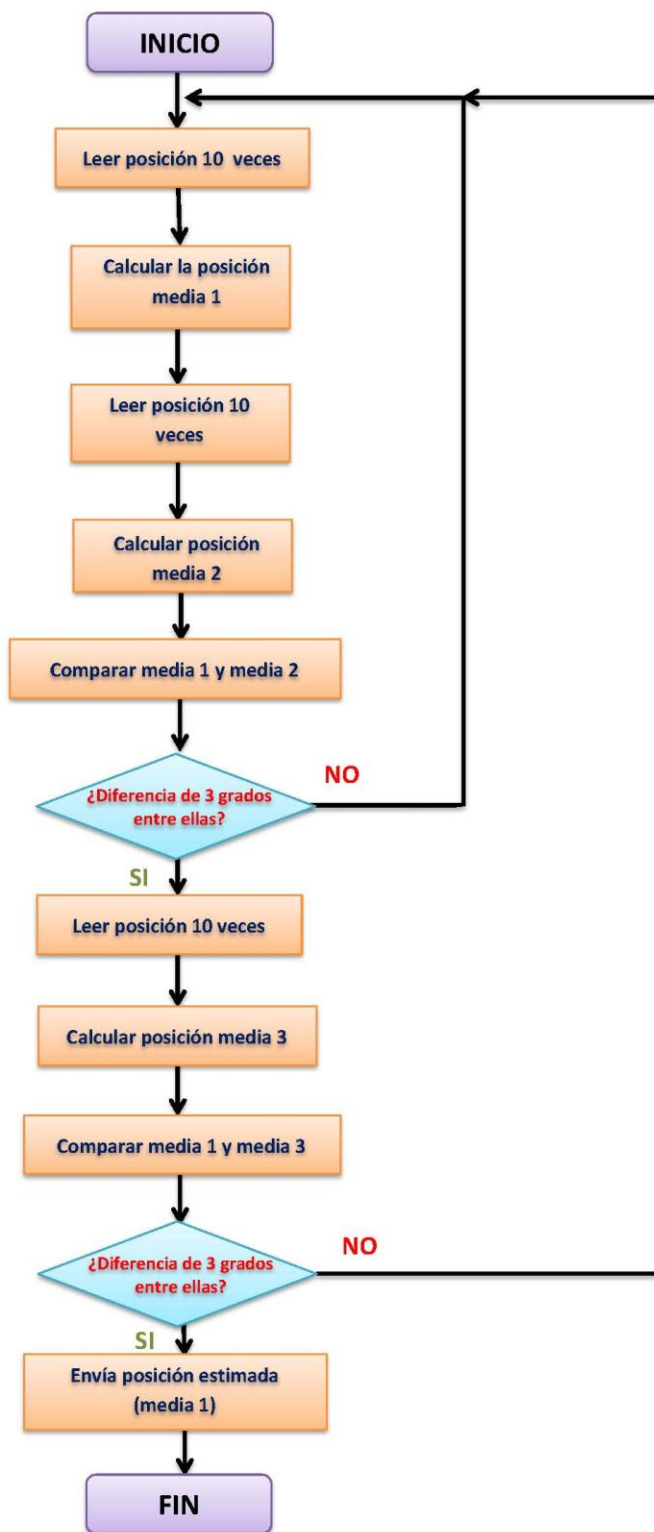


Figura 18 - Flujograma de la lectura estimada con un error de tres grados



Lectura precisa a partir de la estimada con un error de tres grados

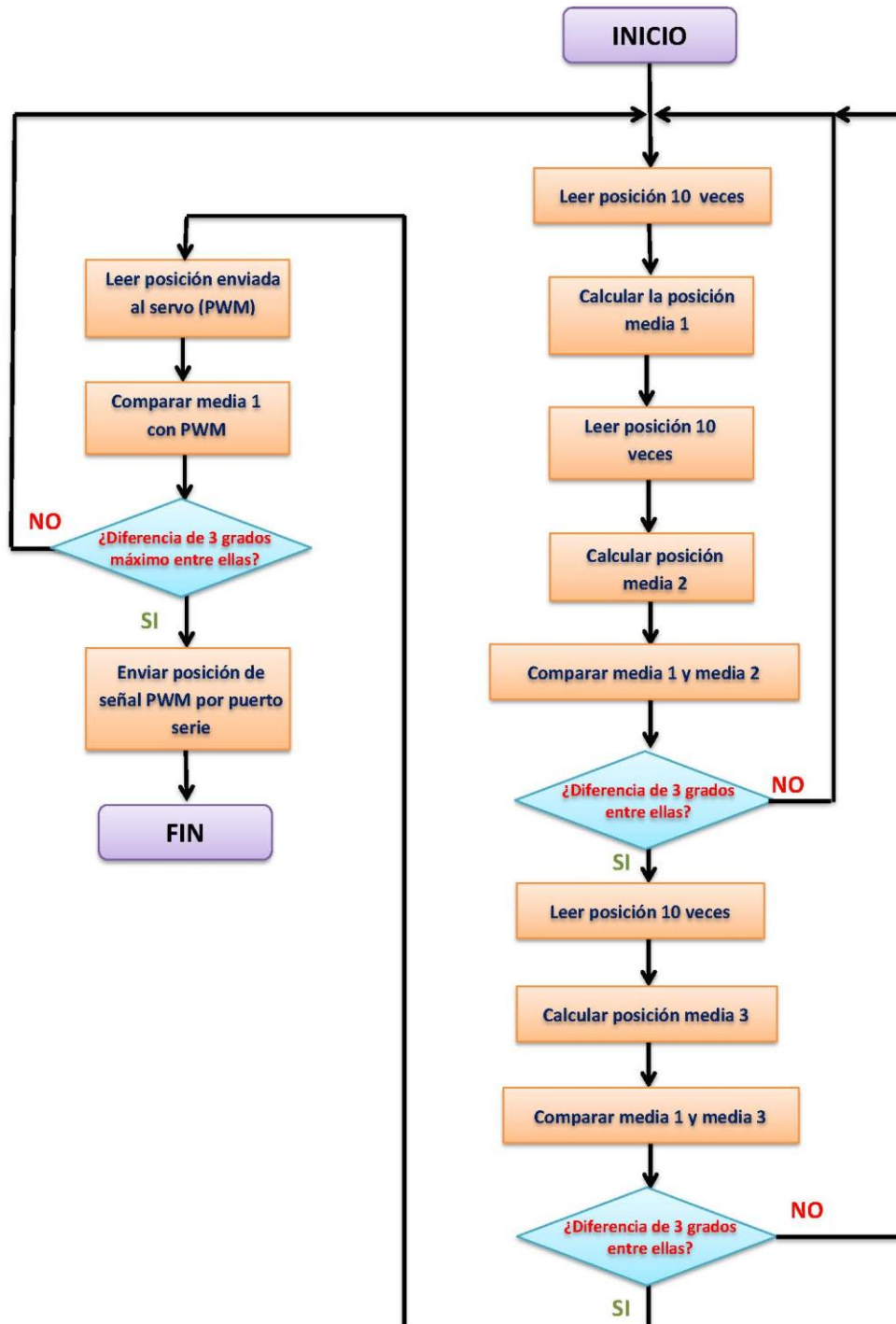


Figura 19 - Flujograma de la lectura precisa a partir de la estimada con un error de tres grados

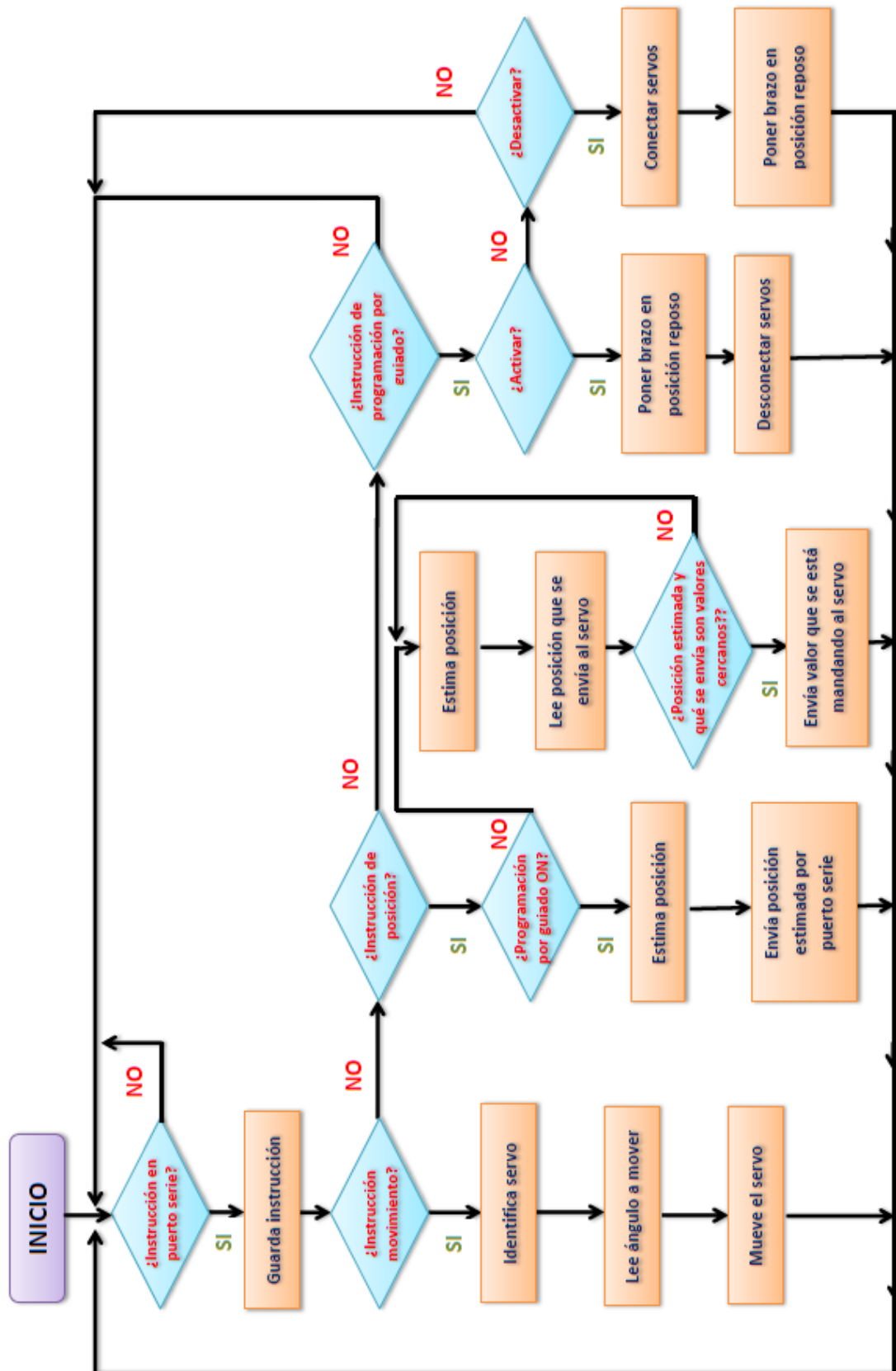


Figura 20 – Flujograma del programa principal



Como ya se indicó anteriormente se han creado una serie de comandos de manera que enviados por el puerto serie se controla el brazo robótico.

Para que el comando enviado sea correcto hay que tener en cuenta lo siguiente:

- Los comandos siempre finalizan con un punto, pues el programa creado lee datos del puerto serie hasta el primer punto.
- Para identificar los servos en los comandos se les ha asignado a cada uno una letra del abecedario, siendo:
  - A -> Servos encargados del movimiento del eslabón 1 (primer grado de libertad).
  - B -> Servos encargados del movimiento del eslabón 2 (segundo grado de libertad).
  - C -> Servo encargado del movimiento del eslabón 3 (tercer grado de libertad).
  - D -> Servo encargado del movimiento del eslabón 4 (cuarto grado de libertad).
  - E -> Servo encargado del movimiento del eslabón 5 (quinto de libertad).
  - F -> Servo encargado del movimiento del eslabón 6 (sexto grado de libertad).
  - G -> Servo encargado del movimiento de la pinza
- El ángulo en los comandos siempre debe tener tres dígitos, completar con ceros las posiciones necesarias hasta completar los 3 dígitos. Debe ser un valor comprendido entre 000° y 180° exceptuando el primer eslabón que puede tomar valores entre 000° y 360°.
- Existen tres tipos de comandos (instrucciones):
  - ➔ **Movimiento (M):** se encarga de mover el servo indicado hacia el ángulo indicado en la instrucción.
  - ➔ **Posición (P):** se encarga de enviar la posición (ángulo) en el que se encuentra el servo indicado en la instrucción por el puerto serie.
  - ➔ **Programación por guiado (PGON ó PGOFF):** se encarga de encender o apagar la programación por guiado. Al activarla deja el brazo en posición de reposo listo para moverlo manualmente permitiendo que adopte la configuración que interese, para obtener el valor del ángulo de cada servo en esa posición concreta. Al desactivarla, el brazo se coloca automáticamente en la posición de reposo, permitiendo moverlo ya con comandos.
  - ➔ La información en los comandos debe enviarse todo junta, sin espacios y en el siguiente orden:

*Letra del servo + instrucción a realizar(M,P,PGON ó PGOFF)+ ángulo de tres dígitos +.*

Nota: si se quiere la posición, el ángulo no hace falta enviarlo, ver ejemplos.

Ejemplos de instrucciones a enviar (PC->Brazo robótico)

MOVIMIENTO:	AM030. BM010. CM120. DM180 ...
PETICIÓN ÁNGULO EN EL QUE ESTÁ:	AP. BP. CP. DP. ...
PROGRAMACIÓN POR GUIADO (ON/OFF):	PGON. PGOFF.

Ejemplos de información a recibir (Brazo robótico->PC)

ENVÍO POSICIÓN ACTUAL DEL SERVO:	AP030. BP010. CP120. DP180. ...
----------------------------------	---------------------------------



# Capítulo 4

## Montaje y cableado

## Capítulo 4: Montaje y cableado

### 4.1. Montaje mecánico y cableado en la estructura

Una vez llegados a este punto está todo listo para comenzar con el montaje del brazo robótico.

Éste capítulo ha sido ideado con la finalidad de tener unas instrucciones gráficas de todo el montaje con la menor redacción posible para una mayor rapidez y un mejor entendimiento.

En cada apartado aparece un recuadro donde se indica la cantidad de piezas que componen el eslabón y los tornillos u otros elementos que se necesitan para el montaje (véase figura 11). Así, antes de comenzar con el montaje del eslabón, se conocerá de antemano si posee todo lo necesario para poder montarlo sin problemas.

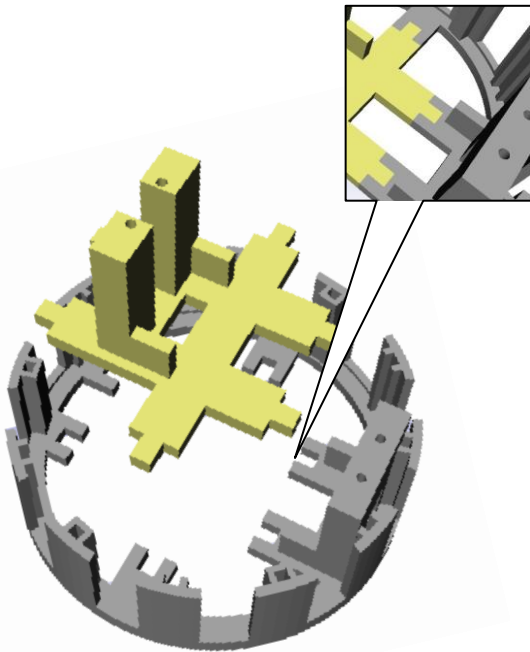
Los apartados están divididos por pasos, las piezas amarillas con las que se van a mover durante ese paso.

Notas:

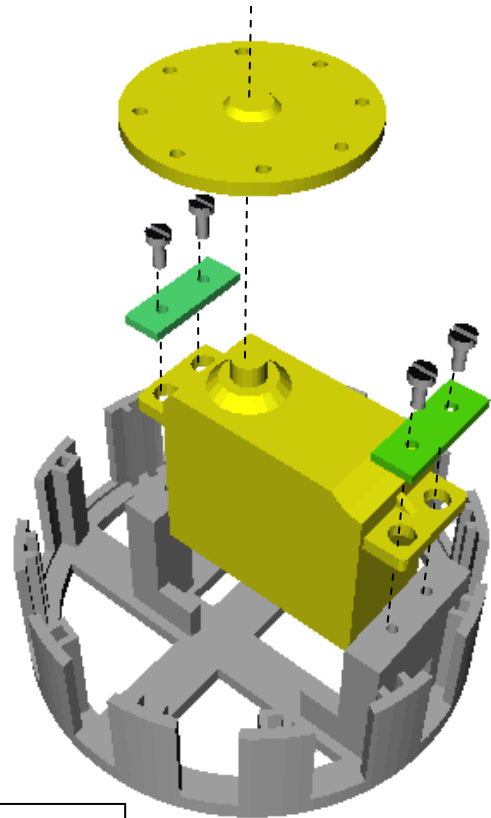
- Tener especial cuidado al montar los servos que mueven el eslabón uno, pues entre los dos deben girar  $360^\circ$
- Todo tornillo que sobresalga demasiado y se prevea que va a dar problemas por su longitud, deberá ser cortado para evitar dichos problemas ó en su defecto poner otro de menor tamaño si se encuentra la medida estándar correspondiente.

### 4.1.1. Eslabón 0

**1**



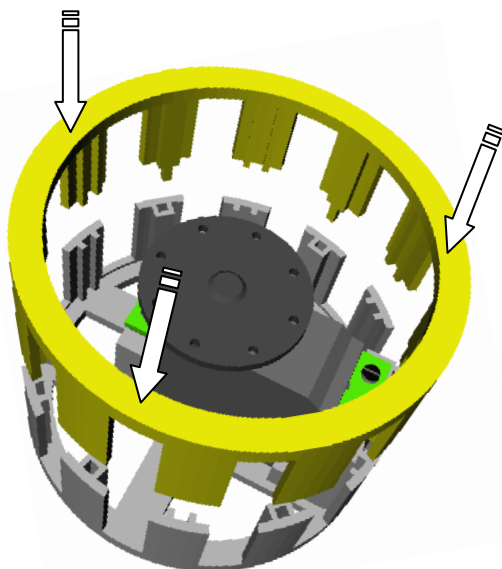
**2**



#### ANOTACIONES:

- Número de piezas: 3
- Servo S3003 , juego completo (tornillos, gomas y plato)
- Pegar las piezas al encajarlas

**3**



**4**

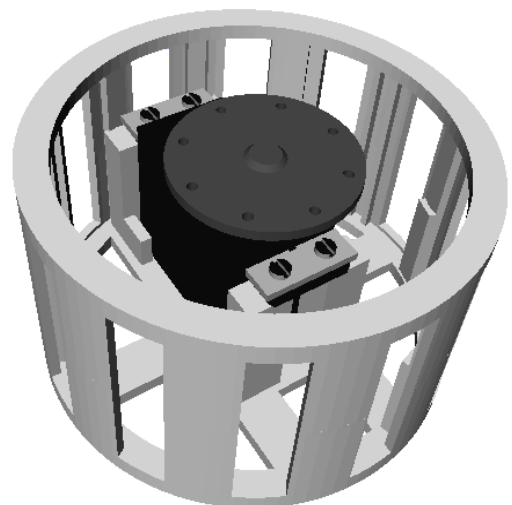
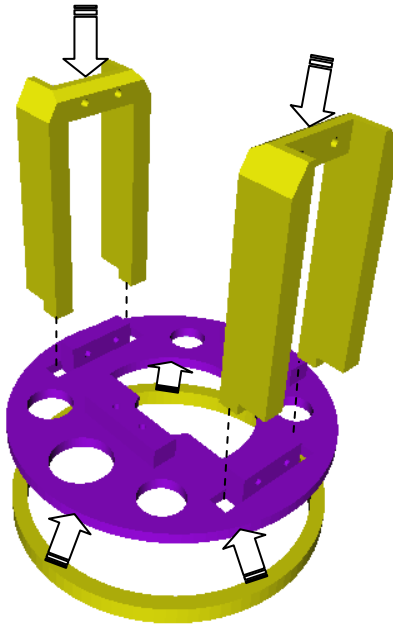


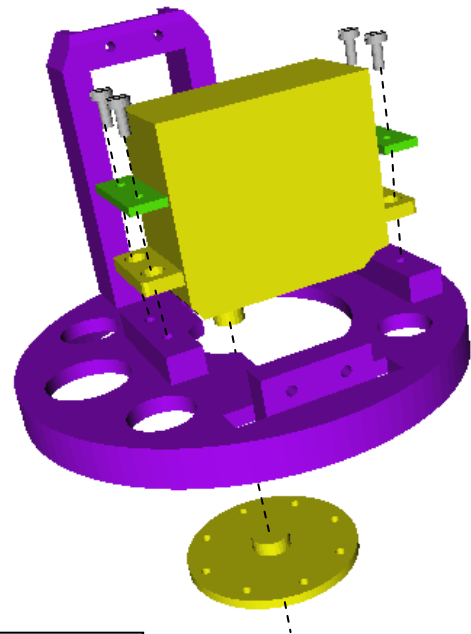
Figura 21 - Montaje de la Base.

### 4.1.2. Eslabón 1

**1**



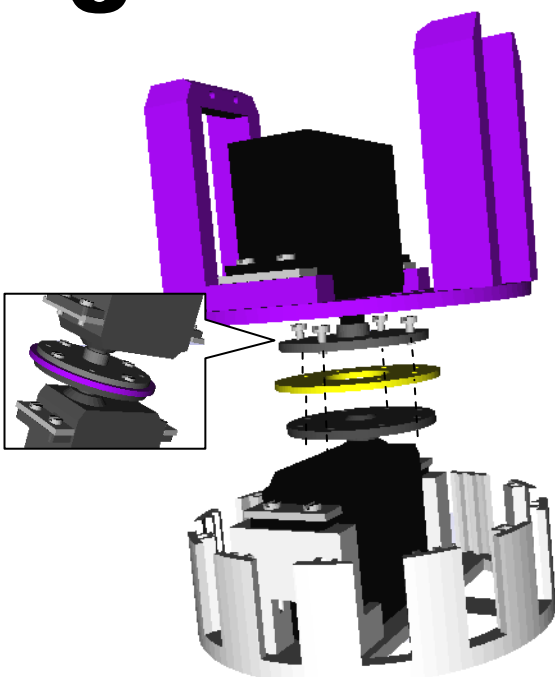
**2**



#### ANOTACIONES:

- Número de piezas: 4
- Servo S3003, juego completo (tornillos, gomas y plato)
- Pegar las piezas moradas al encajarlas.
- 4 tornillos tipo e.

**3**



**4**

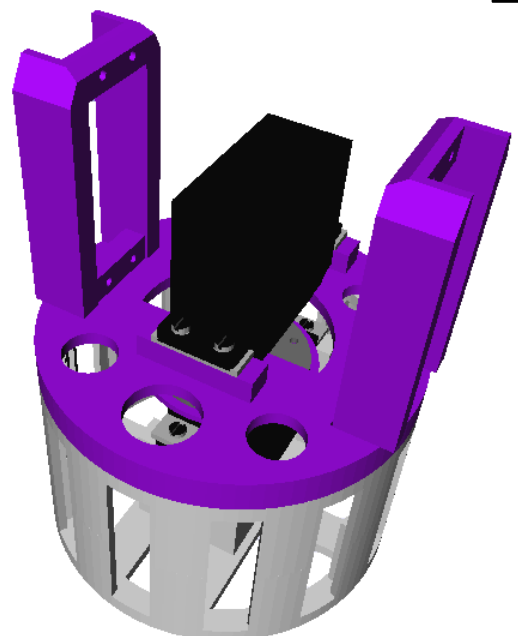
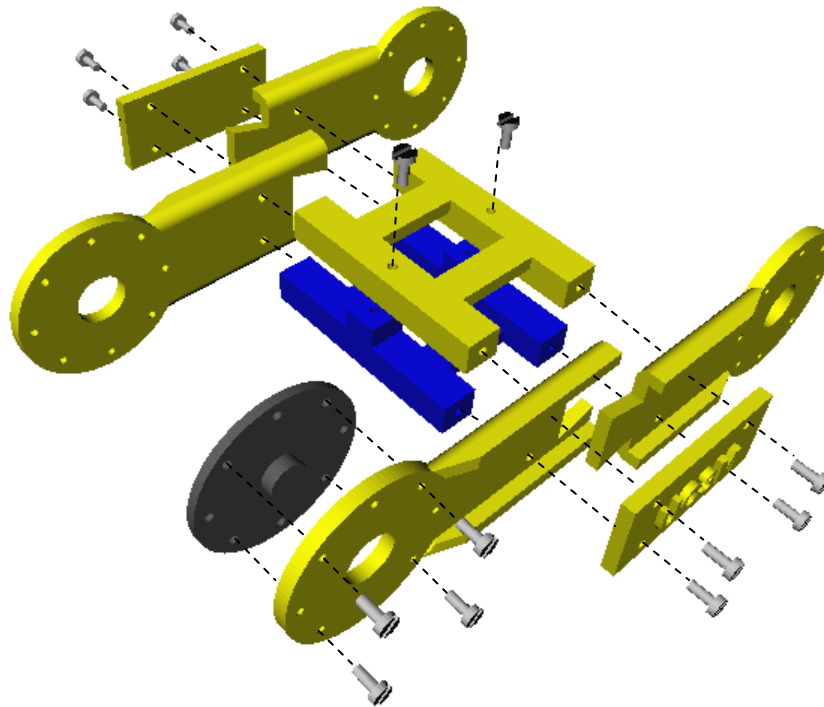


Figura 22 - Montaje eslabón 1.

### 4.1.3. Eslabón 2

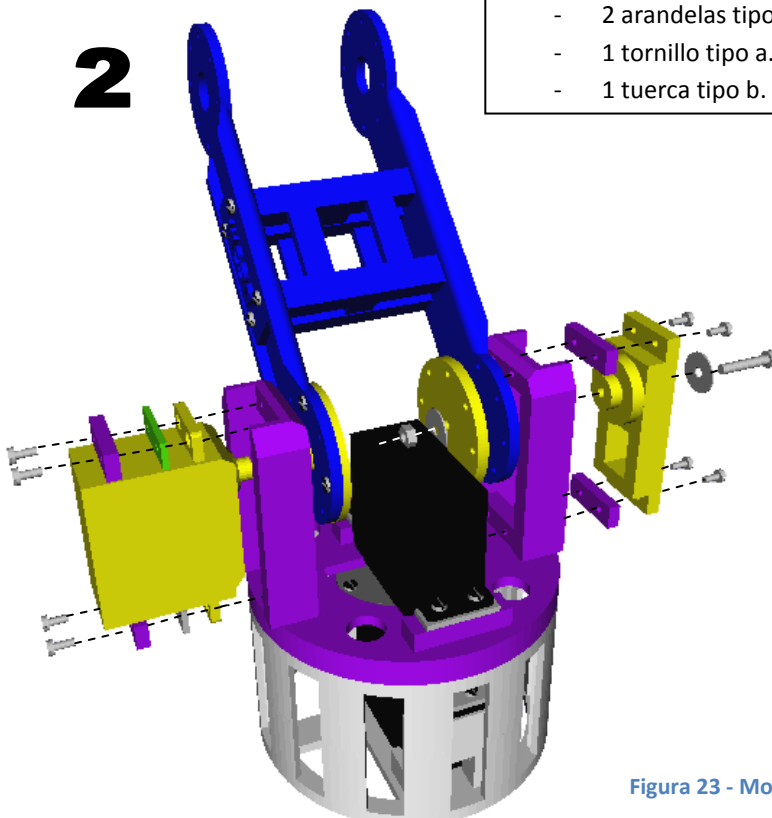
**1**



#### ANOTACIONES:

- Número de piezas: 14
- Servo S3305, juego completo (tornillos, gomas y plato).
- 14 tornillos tipo d.
- 4 tornillos tipo e.
- 2 arandelas tipo c.
- 1 tornillo tipo a.
- 1 tuerca tipo b.

**2**



**3**

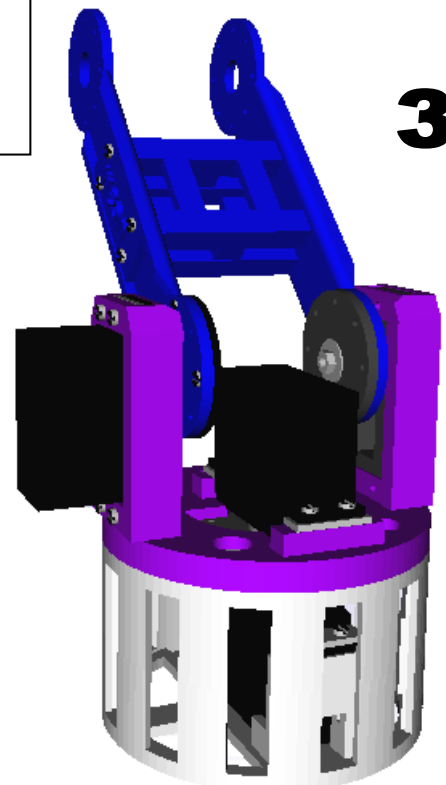
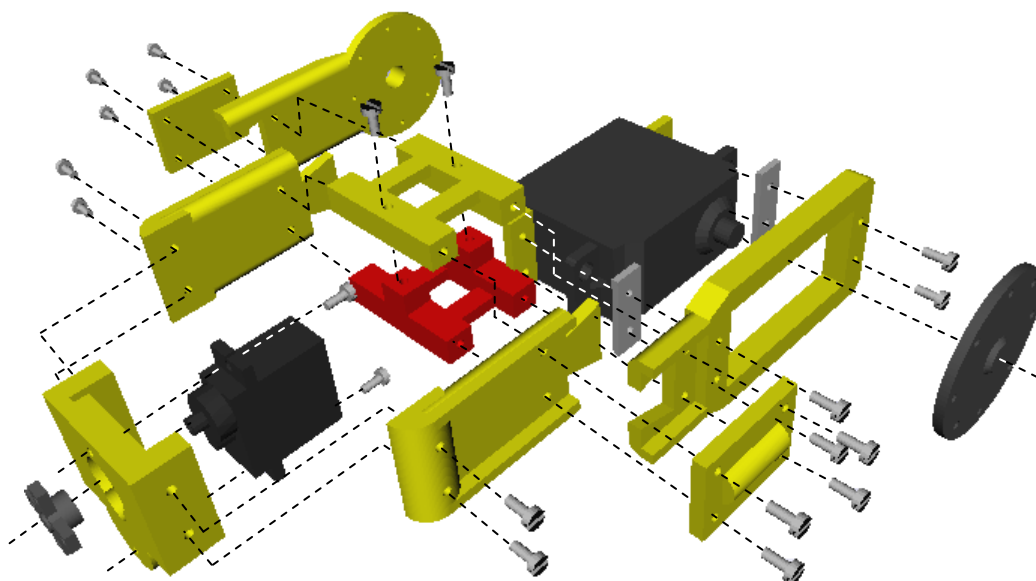


Figura 23 - Montaje eslabón 2.

#### 4.1.4. Eslabón 3

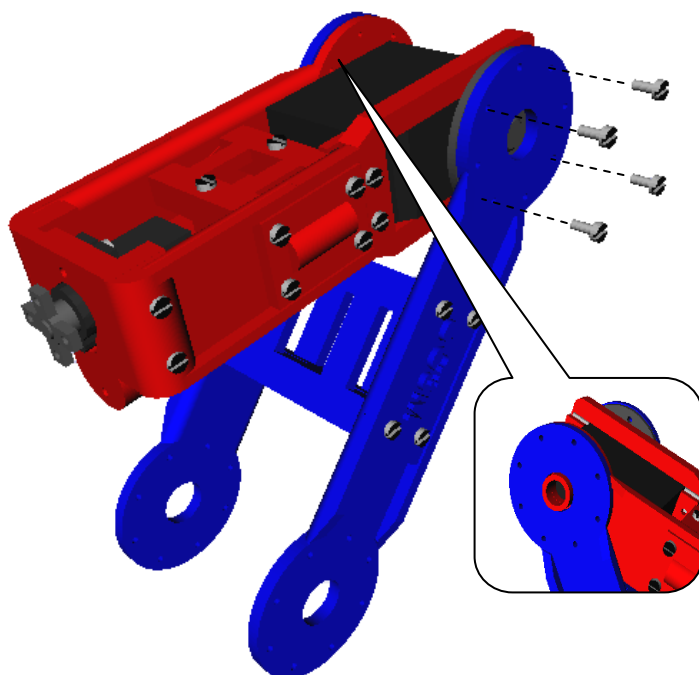
1



##### ANOTACIONES:

- Número de piezas: 11
- Servo S3003, juego completo (tornillos, gomas y plato).
- Servo MG90S, juego completo (tornillos y plato).
- 14 tornillos tipo d.
- 4 tornillos tipo e.

2



3

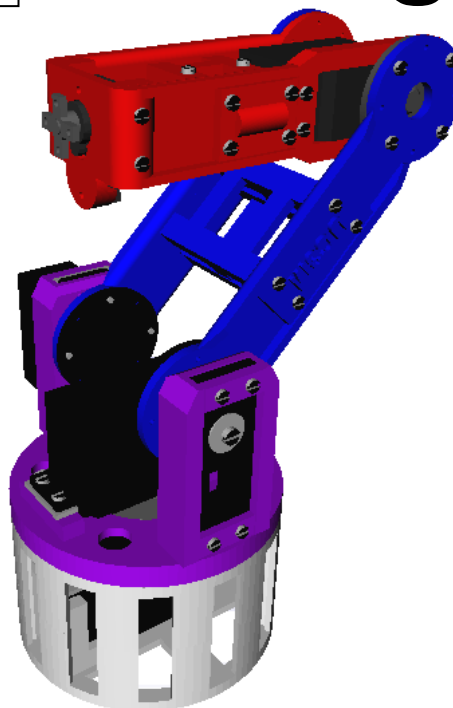
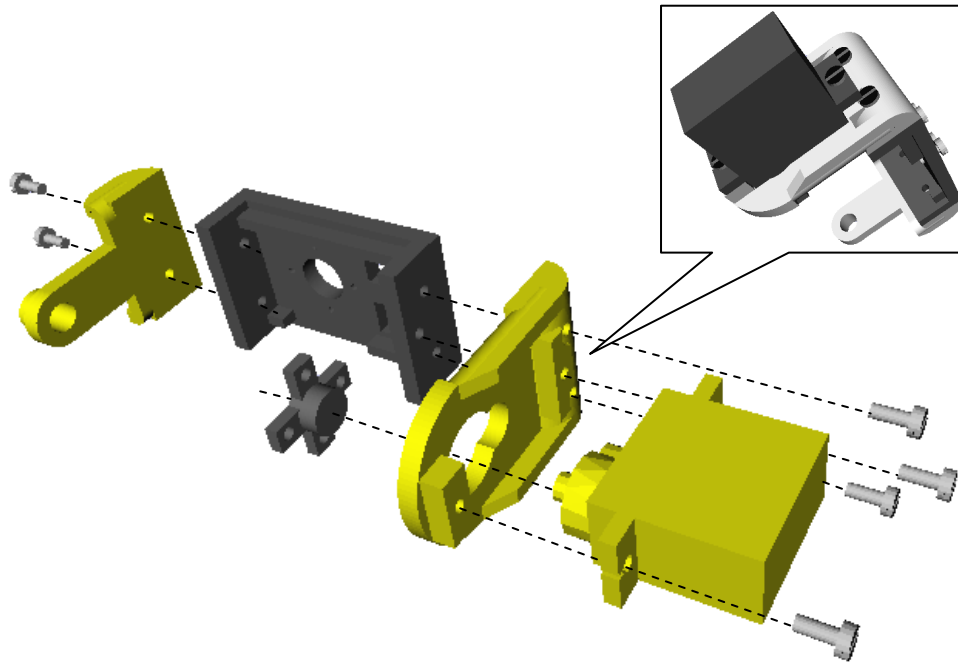


Figura 24 - Montaje eslabón 3.

#### 4.1.5. Eslabón 4

1



**ANOTACIONES:**

- Número de piezas: 3
- Servo MG90S , juego completo (tornillos y plato)
- 4 tornillos tipo e.
- 4 tornillos tipo d.

2

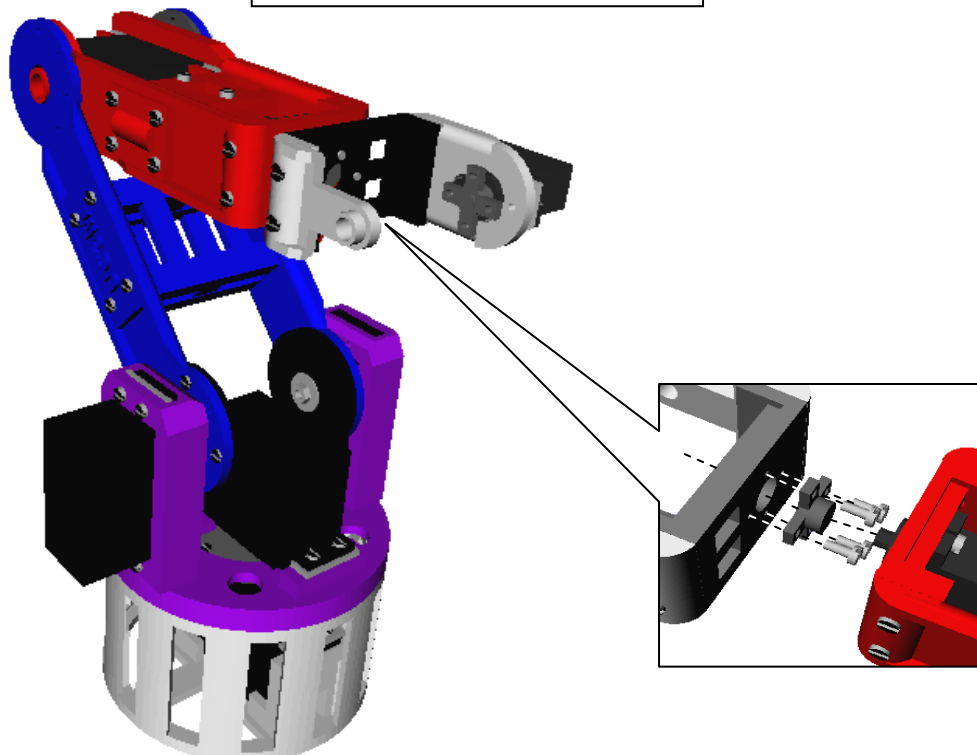
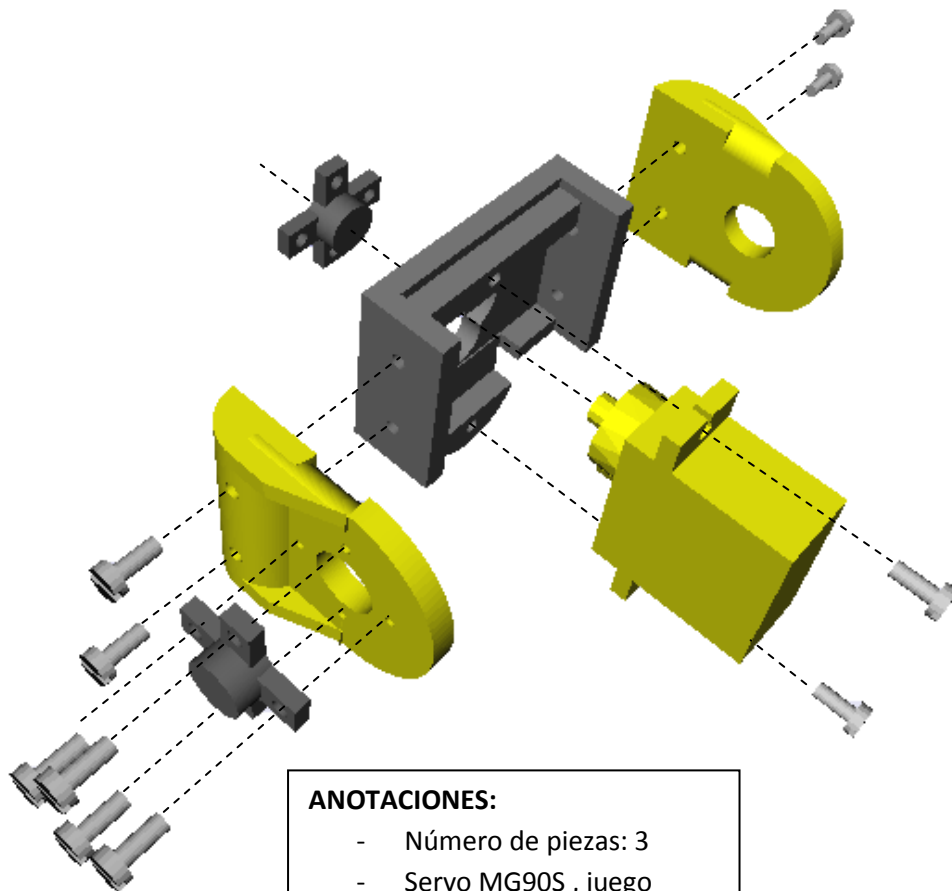


Figura 25 - Montaje eslabón 4.



#### 4.1.6. Eslabón 5 y 6

1



**ANOTACIONES:**

- Número de piezas: 3
- Servo MG90S , juego completo (tornillos y plato)
- 4 tornillos tipo e.
- 4 tornillos tipo d.

2

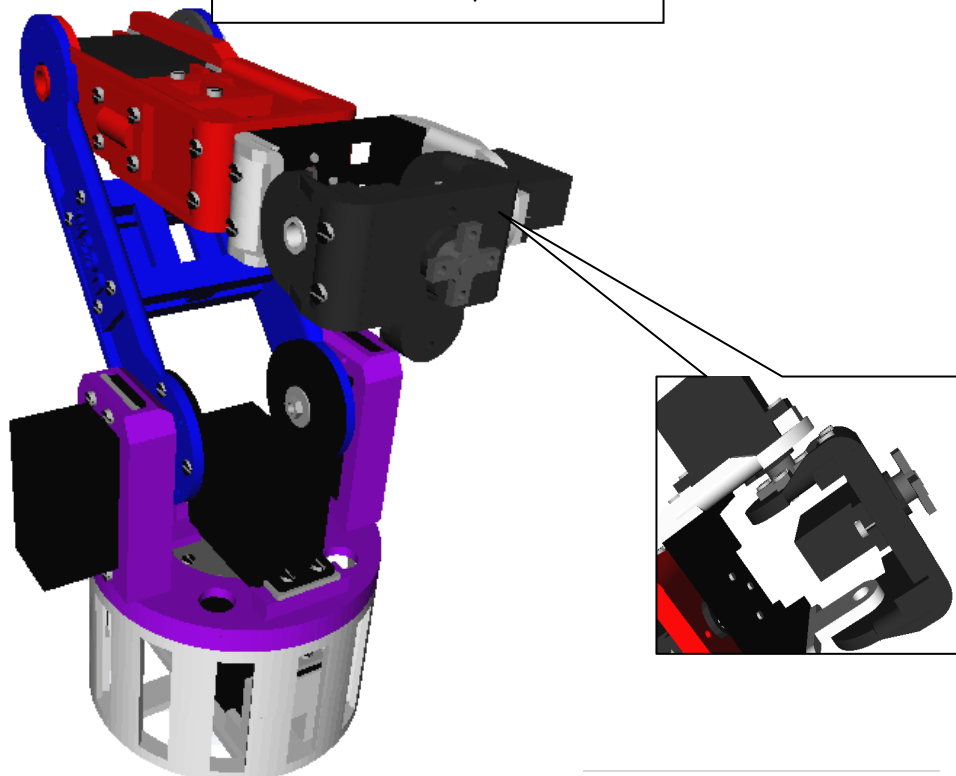
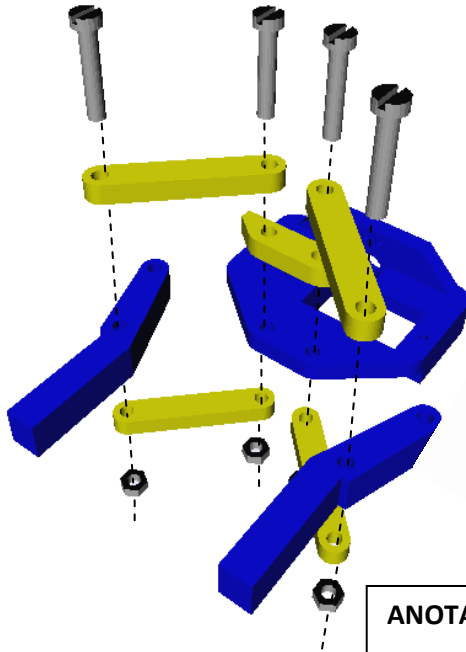


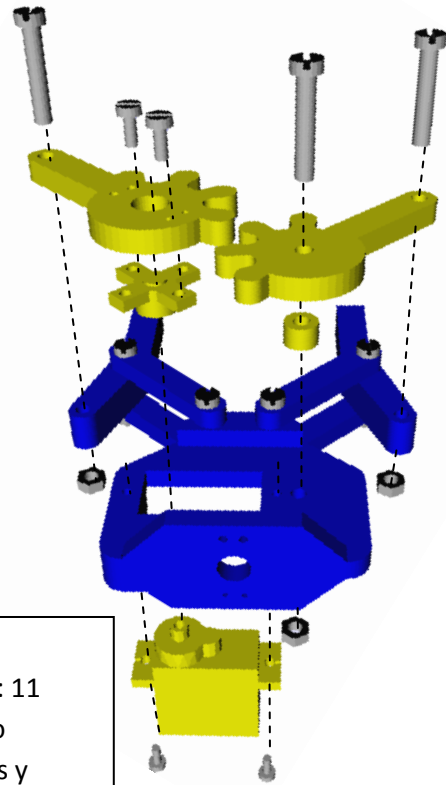
Figura 26 - Montaje eslabón 5.

#### 4.1.7. Pinza

1



2



**ANOTACIONES:**

- Número de piezas: 11
- Servo HS-55, juego completo (tornillos y plato)
- 7 tornillos tipo a.
- 4 tornillos tipo e.
- 7 tuercas tipo b.

3

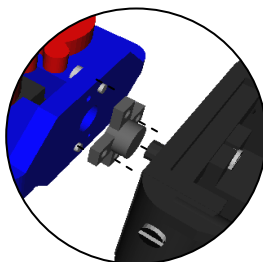
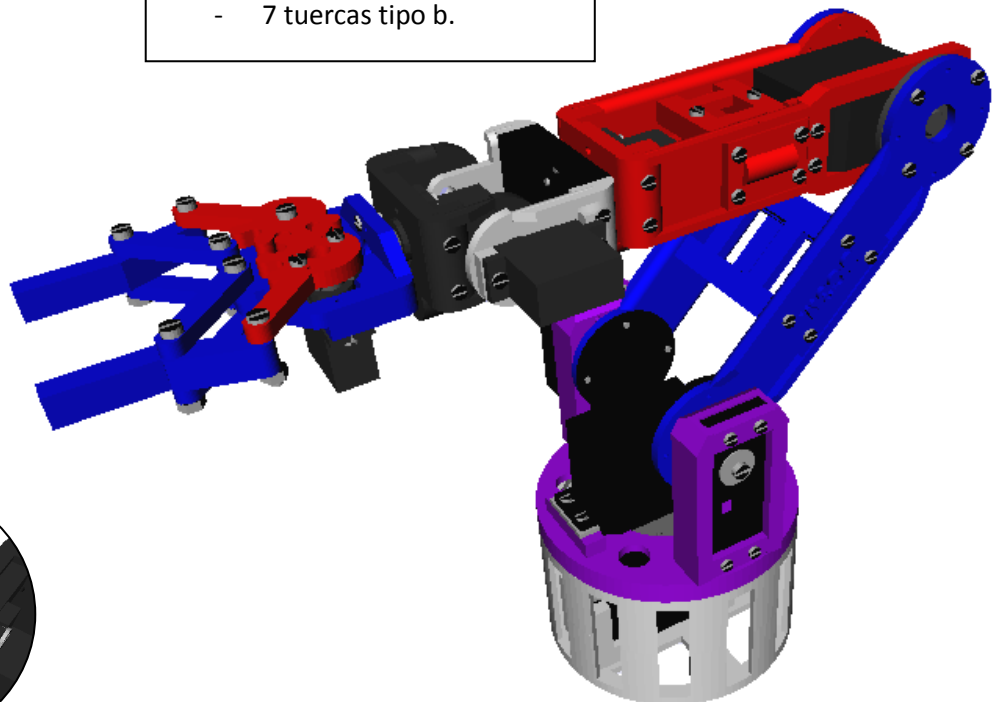


Figura 27 - Montaje de la pinza.

#### 4.1.8. Cableado

En este apartado se va a describir cómo han de colocarse los cables a lo largo de todo el brazo robótico, de manera que no impidan ni limiten el giro en las articulaciones.

Este proceso es de los más complicados, pues la funcionalidad del brazo depende de la colocación que se le dé a los cables.

En la figura 28 se puede observar todas las partes del brazo robótico ya montado, cableado y funcionando. Esta imagen será de gran ayuda durante el cableado.



Figura 28 - Perfil izquierdo, parte frontal, parte trasera, perfil derecho del brazo.

Para el montaje se va a necesitar el siguiente material:

- Bridas pequeñas.
- Manguera recoge cables.
- Cinta aislante.
- Soldador y estaño
- Tubo termoretráctil.
- Alicates de corte.

Antes de empezar a fijar los cables, se va a poner en el conector de los servos que deben estar conectados a su respectivo prolongador, una vuelta de cinta aislante. De este modo, se asegura que durante el montaje y durante el movimiento del brazo, no se desconecta (si tuviera algo de holgura) ningún prolongador.

Los servos serán nombrado como en los capítulos anteriores (de la A hasta la G). Se comenzará el montaje por el servo de la pinza, servo G.

Tanto el módulo central del eslabón dos como del eslabón tres se componen de dos piezas. Quitar los tornillos de la superficie y lo de los laterales, de forma que se pueda quitar la “tapa” para camuflar en su interior los cables (ver figura 29).

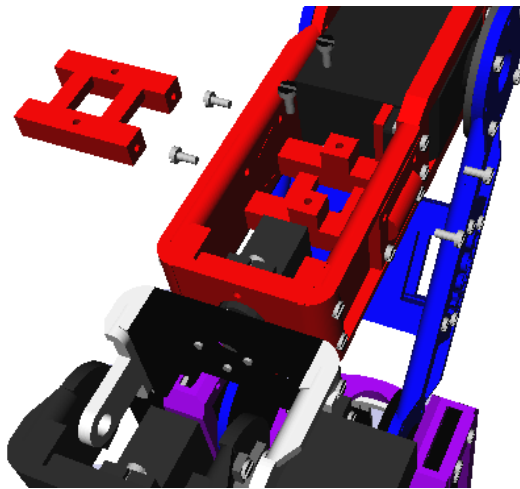


Figura 29 - “Destape” eslabón 3 (análogo para eslabón 2)

Nota1: al referirse en las instrucciones al cable del servo, se incluye al cable de posición también, cualquier excepción se indicará en los pasos de cableado.

Nota2: para facilitar el cableado, se van a formar a lo largo del brazo dos mangueras de cable, colocando ambas mangueras cubriendo la superficie del brazo para equilibrar peso. Una manguera corresponde a los servos G y E y la otra a los servos D y F.



#### Pasos a seguir:

- Coger cable del servo G y unirlo con una brida al servo E por debajo. Comprobar que se ha dejado suficiente cable antes de fijarlo firmemente para que la pinza gire libremente en todo su recorrido.
- Coger cable servo G y E y atarlos al lateral izquierdo (vista frontal) del servo D, pasando los cables de ambos servos por el lado izquierdo interior del modulo central del eslabón tres, dejando el conector de serie y del prolongador del servo G en el interior del módulo central. Dejar que los cables de ambos servos caigan por la parte frontal. Comprobar que la mueca con los cables atados es capaz de hacer todos los movimientos.
- Coger cable del servo F y atarlo junto a los cables del servo D, haciéndolos pasar por el lado derecho interior del eslabón tres, dejándolos que caigan por la parte trasera del brazo (caen por el lado contrario a los cables de los servos G y E)
- “Tapar” el módulo central del eslabón tres, poniendo los seis tornillos quitados anteriormente. Observar que los cables de todos los servos y los conectores del servo G se quedan bien colocados en su interior. Cables de G y E por el lado izquierdo (con conectores de G) y cables del F y D por el lado derecho.
- Cortar un trozo de manguera recoge cables y ponerla en los cables de los servos G y E. dejando en el interior de la manguera el conector del servo E. Pasar la manguera por el lado derecho interior del eslabón dos. Dejar caer los cables por la parte frontal. Sujetar la manguera al modulo central del eslabón dos y comprobar que no impide en su movilidad al eslabón tres.

Nótese que la manguera pasa del lado izquierdo del eslabón tres al lado derecho del eslabón dos, quedándose al poner en vertical el brazo, torcida, permitiendo a la articulación girar sin resistencia.(véase figura 30 cómo se dobla la manguera)

- Coger cables de los servos F y D y pasarlos por el lado izquierdo interior del eslabón dos. Dejando los conectores de F y D en su interior del módulo central de dicho eslabón. Dejar caer los cables por la parte frontal. Atarlos al módulo central del eslabón dos y comprobar que la articulación gira sin problemas (a estos cables no se le ponen manguera puesto que el ángulo que queda al doblar el brazo es pequeño en comparación con el ángulo formado al doblar el brazo en el extremo opuesto e impediría el giro completo (véase figura 30).

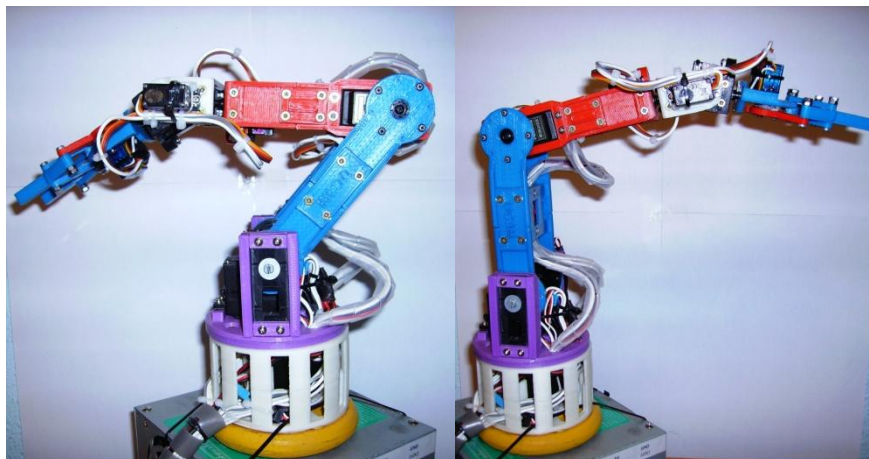


Figura 30 - Ángulo formado entre eslabón dos y tres en posiciones extremas.

- Coger cables del servo C y pasarlos por el lado derecho interior del eslabón dos.
- En los servos, D, E, F, G se separan los cables de alimentación de los cables de señal, se coge un metro de cable bipolar, se sueldan los cinco cables de masa juntos y los cinco cables de tensión juntos. Para evitar cortocircuitos se les pone tubo termoretráctil y se dejan en el interior del módulo central del eslabón dos. De este modo se ha conseguido pasar de ocho cables de alimentación a sólo dos, permitiendo trabajar más cómodamente.
- Cerrar el módulo central del eslabón tres poniéndoles los tornillos quitados anteriormente. Observar que los cables se quedan bien colocados en su interior, tanto los que deben salir fuera del módulo (de posición, PWM, alimentación y prolongador del servo C) como los soldados y los conectores de F y D. Repartir todos los cables que deben salir fuera de manera equilibrada, tratando que salgan repartidos en dos mangueras, una por el lado izquierdo y otra por el lado derecho del módulo central.
- Cortar dos trozos de manguera recoge cables y ponerla a los cables que se han repartido a los lados del módulo central en el punto anterior, e introducir los cables por los agujeros situados al lado de la pieza vertical del eslabón uno. Comprobar antes de fijarlos al eslabón uno con una brida, de que permite girar sin problemas el eslabón.
- Sacar por agujero central el cable del servo C. recoger los cables del servo A2 y C de manera que se queden recogidos con sus conectores delante del servo A2, atándolos con una brida a dicho servo (véase figura 31).

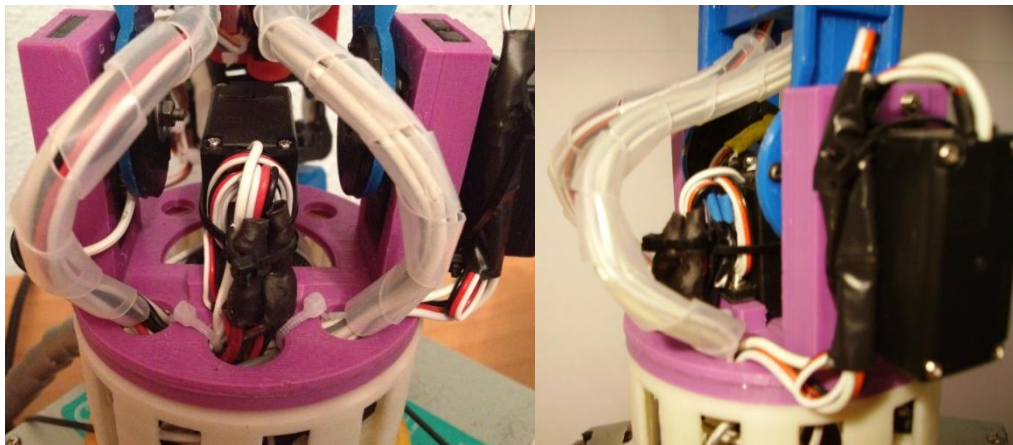


Figura 31 - Colocación de cables servos A1, C (los del interior) y alimentación (los del exterior con tubo termoretráctil negro).

- Recoger cable servo A1 de igual forma que A2 pero fijándolo con una brida a el mismo (servo A1).
- Enrollar el cable del servo B hasta su conector, enrollar cinta aislante alrededor para dejarlo compacto y sujetar el cable con una brida en el mismo, pasando el cable por el agujero de la manguera recoge cables más cercana (véase figura 28 y 31).

- Separar de todos los servos restantes el cable de alimentación y el de señal. Sacar todas las alimentaciones por el agujero que hay entre las dos mangueras y soldarlas sacando de esas soldaduras un cable bipolar de sección más grande, pues por dicho cable se alimentarán todos los servos del brazo. Proteger las soldaduras con tubo termoretráctil y atarlo con una brida al servo A2 (véase figura 31).
- Sacar todos los cables por los agujeros de la base separados del servo, dejando un agujero rectangular entre cada manguera para sacar el cable de posición de los servos A1, A2 y los cables de alimentación (véase figura 32).

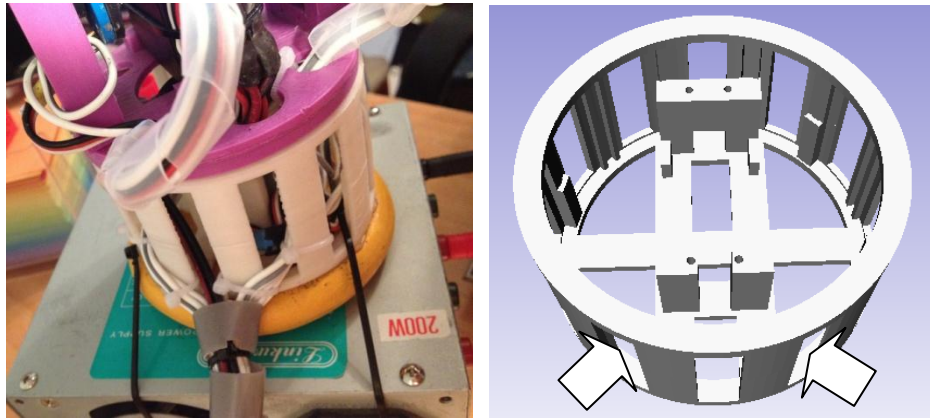


Figura 32 - Colocación de cables en la base I.

Deslizar los cables suavemente hacia dentro, generando una curva, de manera que permita girar al eslabón uno. Comprobar que gira correctamente y poner una brida fijándolos a la base. (ver figura 33).

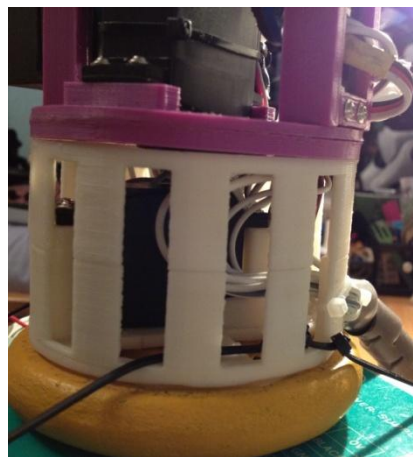


Figura 33 - Colocación de cables en la base II.

- Unir todos los cables creando una manguera solamente e introducirlos por la manguera recoge cables.
- Coger un conector hembra de 10 pines y soldar los cables de señal PWM en el orden marcado por la tabla 7

Pin conector hembra	Servo	Pin Arduino
1	A1	2
2	A2	3
3	B1	4
4	B2 ( brazo 9 servos)	5
5	C	6
6	D	7
7	No conectado	
8	E	8
9	F	9
10	G	10

Tabla 7 - correspondencia de pines y servos al conector hembra del brazo.

- Coger un conector macho de cuatro pines y otro de cinco. Soldar en dichos pines las señales de posición según muestra la tabla 8 y 9.

Pin conector macho (4 pines)	Servo
1	A2
2	B2 (brazo 9 servos)
3	B1
4	A1

Tabla 8 - correspondencia de pines y servos al conector macho de 4 pines.

Pin conector macho (5 pines)	Servo
1	F
2	C
3	D
4	E
5	G

Tabla 9 - correspondencia de pines y servos al conector macho de 5 pines

- Finalmente poner una vuelta de cinta aislante alrededor de los conectores para proteger las soldaduras (véase figura 34).



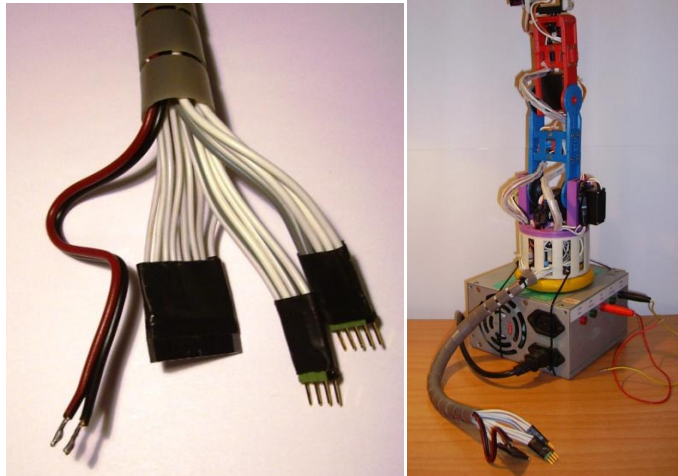


Figura 34 - conectores soldados y brazo robótico cableado

## 4.2. Montaje electrónico y conexinado

### 4.2.1. Montaje placa con electrónica complementaria

Para el montaje de la placa con electrónica complementaria, se necesita los siguientes componentes:

- 9 tiras de 8 conectores macho.
- 1 tira de 6 conectores macho.
- 1 tira de 4, 5, 6 y 8 conectores hembra.
- Dos clemas para PCB.
- Un zócalo de 16 pines (estrecho).
- Multiplexor 74HC4052.
- Soldador y estaño.

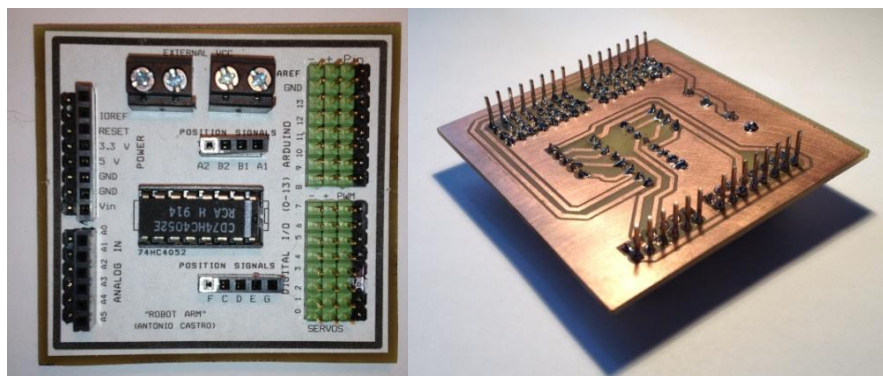


Figura 35 - Placa de electrónica complementaria.

En el CD que se adjunta, se puede encontrar el fotolito para la realización de la placa.

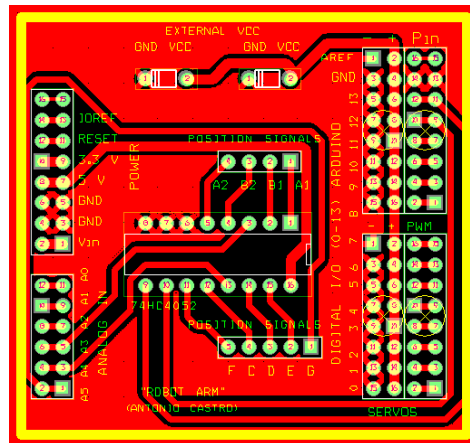


Figura 36 - Diseño de placa de electrónica complementaria.

Para el montaje de la placa hay que tener especial cuidado en no calentar demasiado la placa con el soldador, si sucediera se podría dañar la pista calentada.

Pasos para el montaje (véase como ayuda figura 35 y 36):

- Soldar primero las tiras de conectores macho bocabajo (extremos de la placa), estas tiras son las que conectarán la placa con Arduino UNO.  
Dos tira de 8 conectores macho en “Digital I/O”, una tira de 8 conectores en “Power” y una tira de 6 conectores en “Analog in”.
- Soldar las 6 tiras de 8 conectores macho, en posición normal donde pone “Digital I/O”
- Soldar en “Power” la tira de 8 conectores hembra.
- Soldar en “Analog in” la tira de 6 conectores hembra.
- Soldar el zócalo del multiplexor (centro de la placa).
- Soldar conectores hembra para las señales de posición (tiras de 4 y 5 conectores hembra), se encuentra situadas las tira de conectores una a cada un lado del multiplexor.
- Soldar las dos clemas para PCB.

#### 4.2.2. Conexión de placa complementaria y conexionado del cableado del brazo con Arduino UNO.

La placa de electrónica complementaria se conecta a Arduino UNO, conectando los pines machos boca abajo, sobre los pines hembra de Arduino (véase figura 33 y 35)

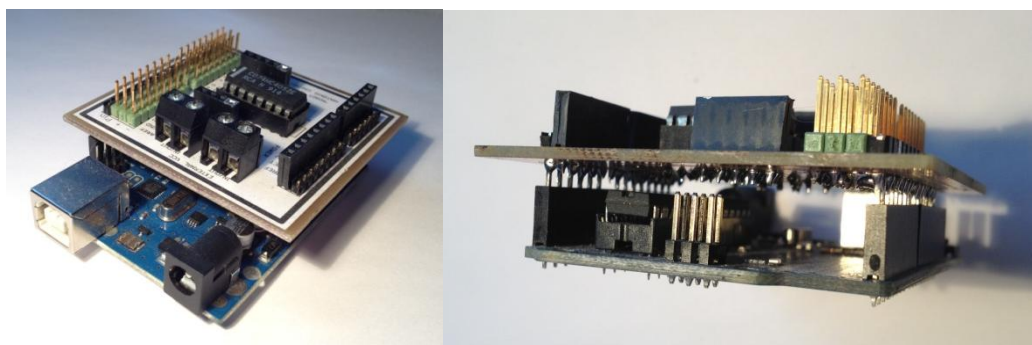


Figura 37 - conexión de placa complementaria y Arduino UNO.

Para los conectores del brazo con la placa, hay que tener especial cuidado en conectarlos de la forma correcta, por ello, como se puede apreciar en la figura 38 se han marcado con un rotulador tanto en la placa como en los conectores del brazo, el primer pin de cada conector evitando así una conexión errónea.

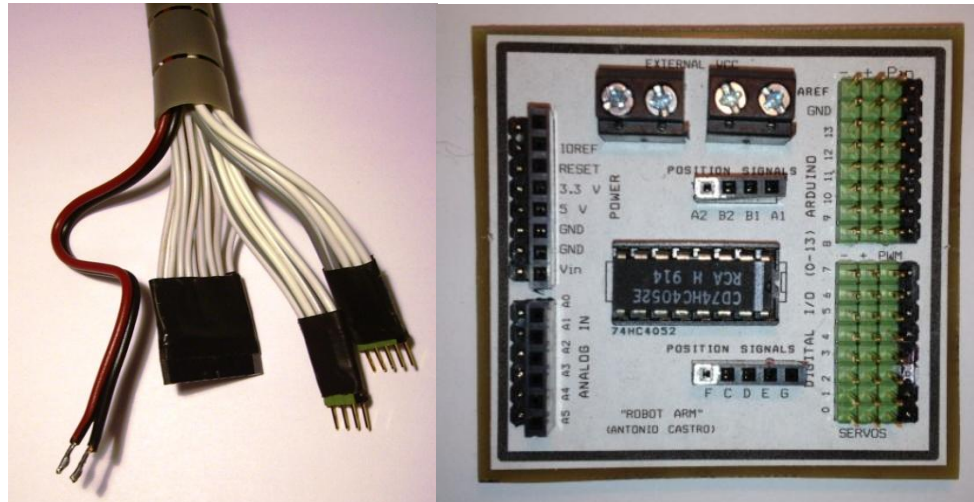


Figura 38 - conectores y placa de electrónica complementaria.

Los conectores macho del robot, pertenecen a las señales de posición mientras que el conector hembra pertenece a la señal PWM de los servos.

La alimentación del brazo, se puede conectar a cualquiera de las dos clemas, ya que en una se conecta la fuente y en la otra el brazo.

La placa conectada queda como aparece en la figura 39.

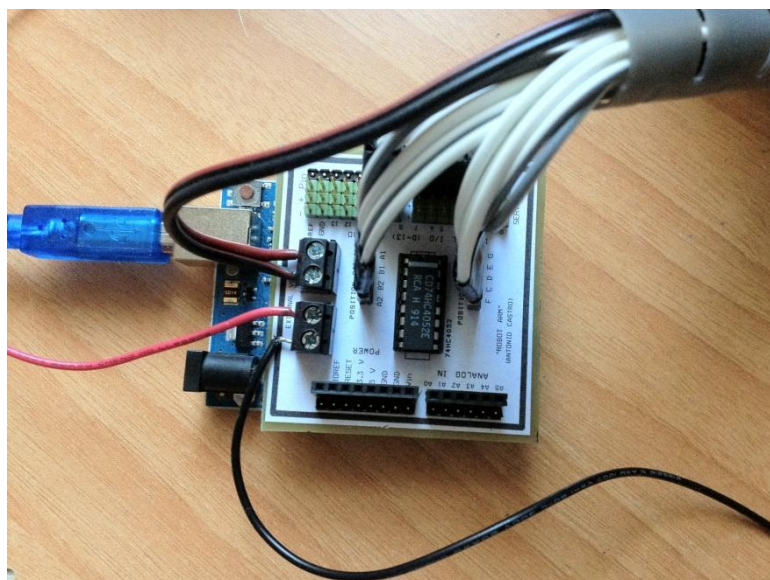


Figura 39 - Placas conectadas al brazo, al puerto serie (USB) y a la fuente de alimentación.

Para conocer la correspondencia entre de los pines de Arduino UNO con los servos véase tabla 7.

### 4.3. Puesta a punto

Una vez que esta todo conectado, es necesario calibrar los servos para tener una lectura de la posición correcta.

Para calibrar el brazo es necesario saber el valor de tensión en las posiciones extremas de cada servo, ya que una vez que se conozca el rango de valores que puede tomar dicho servomotor, siendo el valor mínimo el correspondiente a cero grados y el valor máximo el correspondiente a ciento ochenta grados, se podría conocer el ángulo equivalente para cualquier valor de posición en la que se encuentre el servo.

El convertidor AD de Arduino, tiene una resolución de 10 bits. Los valores de tensión que podemos leer para los que viene configurada la placa por defecto, abarcan el rango de 0 a 5V, por lo tanto cualquier tensión comprendida entre esos valores nos va a dar el convertidor un valor entre 0 y 1024, siendo 0V el valor 0 y 5V el valor 1024.

La ecuación del convertidor es por lo tanto :

$$valor = 204.8 * Tensión$$

El valor que se obtenga de la ecuación anterior será el que hay que introducir en el código del programa.

Para la calibración se va a necesitar:

- Tener todo el brazo conectado a las placas y al PC.
- Un polímetro para medir tensiones.
- El software Terminal 1.9b (para controlar el puerto serie, envío y recepción (lectura))
- El software de Arduino
- El código BRAZO\_ROBOTICO\_CALIBRACION (Anexo II)

Los pasos a seguir para calibrar el brazo son los siguientes:

- Abrimos el programa BRAZO\_ROBOTICO\_CALIBRACION con Arduino y lo descargamos en el microcontrolador. (Este programa se encarga de configurar el brazo robótico para que lea ejecute y envíe información por el puerto serie, además de estimar la posición con los valores de calibración introducidos).
- Ejecutamos el programa Terminal 1.9b y configuramos los parámetros de la comunicación tal y como aparece en la elipse roja de la figura 40. La elipse verde es la zona donde aparecerá lo que se recibe por el puerto serie y la elipse azul es donde se escriben las instrucciones que se envían.



Nota: si Terminal v1.9b está conectado al puerto serie, el programa de Arduino no dejara enviar datos al mismo (como descargar el programa) , por lo tanto hay que desconectarlo , enviar los datos y después volverlo a conectar.

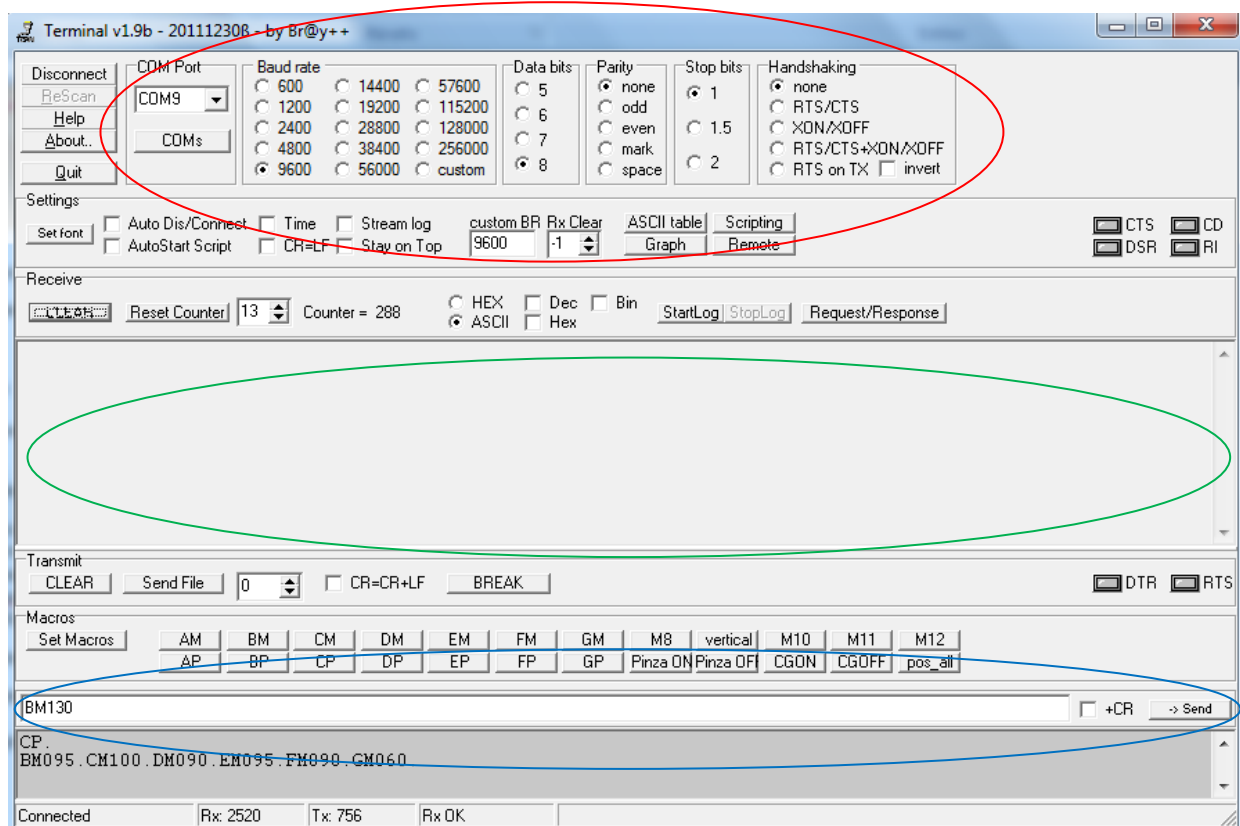


Figura 40 - Terminal v1.9b.

- Se pincha en el botón de arriba la izquierda y se le da a “conectar”.
- Se envía por el puerto serie una instrucción de movimiento que lleve al servo indicado a la posición de 0°.
- Con el polímetro se mide en la placa la tensión continua que tiene el servo en esa posición.
- Mediante la ecuación del convertidor se calcula el valor.
- El valor calculado se introduce dentro del código de programa, en la parte que dice “calibración” y en su línea de código correspondiente.

Si la articulación tiene dos servos en la instrucción se indica si es el servo A o el B.

#### Ejemplo 1:

Si se ha calculado el valor para 0° del servo A2 y ha sido de 97, la línea de código sería:

`int calE1B0=97;`

donde cal=calibración, E1= eslabón uno, B=servo B por ser A2, 0=0°.

Si la articulación solo tiene un servo, la instrucción sería la misma pero quitando la letra.

Ejemplo 2:

Calculado el valor para 180° del servo C y ha sido de 450, la línea de código sería:

`int calE3180=450;`

donde cal=calibración, E3= eslabón tres, 180=180°.

- Se repite el proceso para el mismo servo pero esta vez llevándolo a la posición de 180°.
- Volver a realizar todos los pasos para el resto de servos.

Una vez que se haya realizado los pasos anteriores para todos los servos, habiendo introducido en el código del programa los valores calculados, hay que comprobar si verdaderamente lee bien el brazo, para ello se va moviendo el servo en posiciones aleatorias (ejemplo: *AM130.*), una vez movido el servo leemos su posición (ejemplo: *AP.*) el brazo nos debería enviar la posición y si todo es correcto, nos debería mandar la posición a la que se le dijo que fuera (en el ejemplo 130°) si la posición difiere y se desea ajustar mas, hay que ir modificando los valores para 0° y 180° del servomotor a ojo, hasta dar con los valores que mejor den la posición.



# Capítulo 5

## Modelo Cinemático

## Capítulo 5: Modelo Cinemático

El modelo cinemático que ahora presentamos ha sido realizado por Carlos Rodríguez Zambrana para su TFG titulado “Modelo cinemático y control de un brazo robótico imprimible”, utilizando el prototipo de brazo robótico diseñado en éste proyecto.

### 5.1. Cinemática

En este apartado analizaremos el movimiento del robot con respecto a un sistema de referencia situado en la base. Obtendremos una descripción analítica del movimiento espacial y, en particular, de la posición y orientación del extremo final del robot.

Tenemos dos problemas a resolver en cuanto a la cinemática del brazo robótico:

- **Cinemática directa:** determinar la posición y orientación del extremo final del robot, con respecto a un sistema de coordenadas de referencia, conocidos los valores de las articulaciones.
- **Cinemática inversa:** determinar la configuración que debe adoptar el robot para una posición y orientación del extremo conocidas.

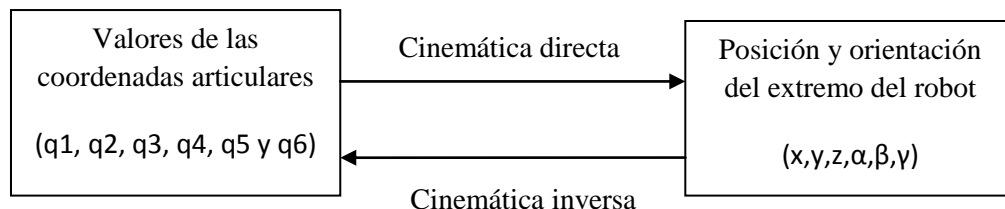


Figura 41 - Diagrama de relación entre cinemática inversa y directa.

Para solucionar el primer problema se utilizará el Algoritmo de Denavit-Hartenberg. De esta forma, se obtiene la posición del extremo del robot a partir de los valores de los ángulos del mismo.

Para solucionar el problema de la cinemática inversa se ha optado por el método de la matriz de transformación homogénea. Así, se puede determinar los diferentes valores de los ángulos de los ejes del robot para conseguir posicionar su extremo en un punto del espacio establecido por el usuario.



## 5.2. Cinemática Directa

Como se ha explicado anteriormente, la cinemática directa consiste en obtener la posición del robot conociendo los valores de los diferentes ángulos de los ejes del mismo. Para conseguir dicho objetivo se ha utilizado el algoritmo de Denavit-Hartenberg (D-H).

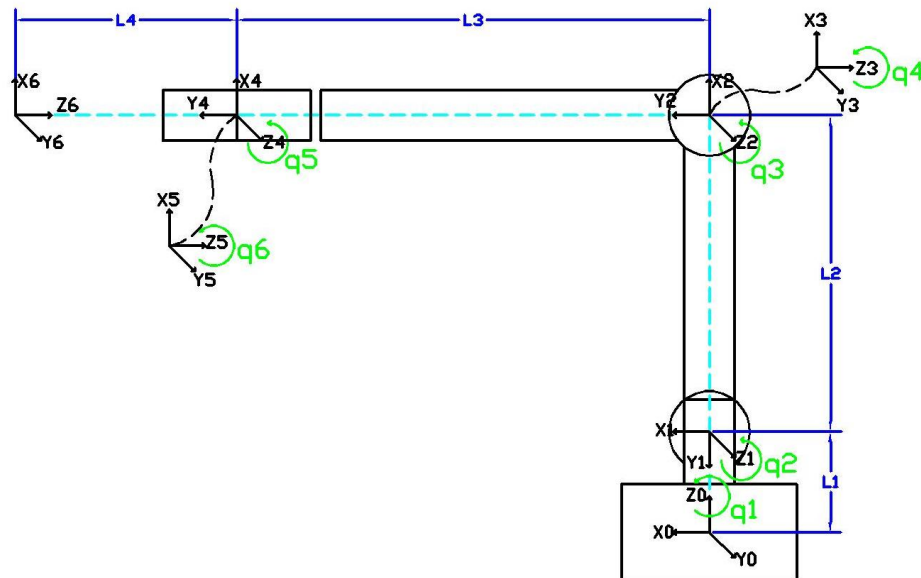


Figura 42 - Diagrama con los sistemas de referencia, eslabones y ejes del robot

A continuación se muestra un diagrama esquemático de los diferentes sistemas de referencia y los ejes del brazo robótico además de la distancia entre eslabones.

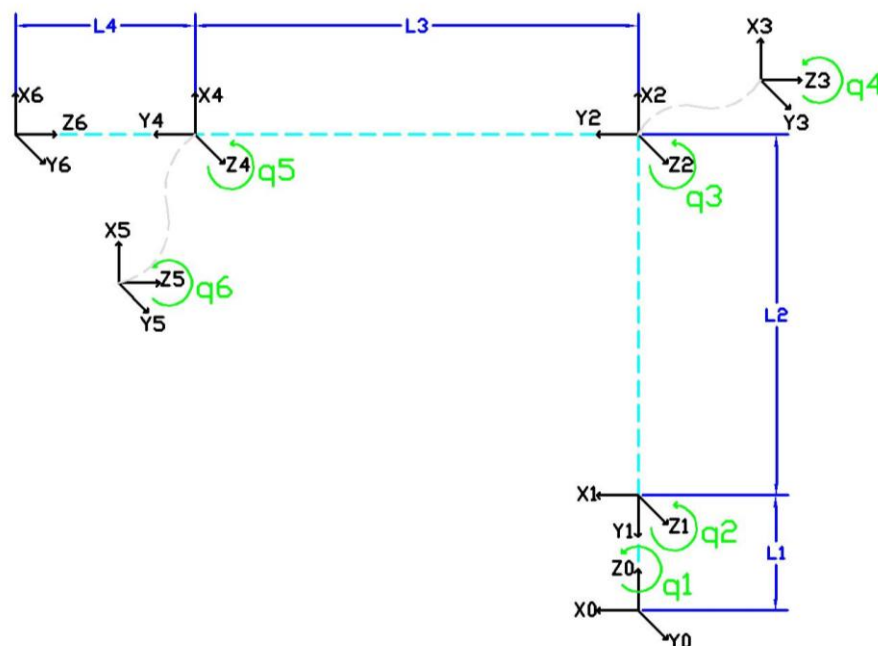


Figura 43 - Diagrama con los sistemas de referencia y los ejes del robot

Siguiendo los pasos marcados por el Algoritmo de Denavit-Hartenberg se ha calculado los parámetros correspondientes y se disponen en la tabla siguiente.

Articulación	$\theta$	d	a	$\alpha$
1	q1	L1	0	90
2	q2-90	0	L2	0
3	q3	0	0	-90
4	q4	-L3	0	90
5	q5	0	0	-90
6	q6	-L4	0	0

Tabla 10 - Parámetros Denavit-Hartenberg

Para la obtención de los parámetros se ha tenido en cuenta que:

- El parámetro  $\theta_i$  es el ángulo que hay que girar sobre el eje  $z_{i-1}$  para que  $x_{i-1}$  y  $x_i$  queden paralelos.
- El parámetro  $d_i$  es la distancia sobre el eje  $z_{i-1}$  que hay que desplazar el sistema  $i-1$  para que  $x_{i-1}$  y  $x_i$  queden alineados.
- El parámetro  $a_i$  es la distancia sobre el eje  $x_i$  que hay que desplazar el sistema  $i-1$  para que su origen coincida con el sistema  $i$ .
- El parámetro  $\alpha_i$  es el ángulo que hay que girar sobre  $x_i$  para que el sistema  $i-1$  coincida con el sistema  $i$ .
- Una vez hallados los parámetros se obtienen las matrices de transformación como expresa la siguiente ecuación:

Ecuación 1. Obtención de la Matriz de transformación D-H

$${}^{i-1}A_{i-1} = RotZ(\theta_i) \cdot Tras([0,0,d_i]) \cdot Tras([a_i,0,0]) \cdot RotX(\alpha_i)$$

Para mayor simplicidad en la notación, el  $\cos(\theta_i)$  está representado por  $C_i$  y el  $\sin(\theta_i)$  está representado por  $S_i$ . La expresión anterior (ecuación 1) se puede convertir en una única matriz quedando representada de la siguiente forma:

Ecuación 2. Matriz de transformación D-H

$${}^{i-1}A_{i-1} = \begin{pmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Ahora, una vez calculados los parámetros de Denavit-Hartenberg, podemos calcular las matrices de transformación de un sistema a otro. A continuación se muestran los resultados obtenidos:

**Ecuación 3. Resultados obtenidos de la Matrices de transformación ( ${}^{i-1}A_i$ ).**

$${}^0A_1 = \begin{pmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad {}^1A_2 = \begin{pmatrix} S_2 & -C_2 & 0 & L_2C_2 \\ C_2 & S_2 & 0 & L_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^2A_3 = \begin{pmatrix} C_3 & 0 & -S_3 & 0 \\ S_3 & 0 & C_3 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad {}^3A_4 = \begin{pmatrix} C_4 & 0 & S_4 & 0 \\ S_4 & 0 & -C_4 & 0 \\ 0 & 1 & 0 & -L_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^4A_5 = \begin{pmatrix} C_5 & 0 & -S_5 & 0 \\ S_5 & 0 & C_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad {}^5A_6 = \begin{pmatrix} C_6 & -S_6 & 0 & -L_4C_6 \\ S_6 & C_6 & 0 & -L_4S_6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Para obtener la matriz de transformación (T) entre la base y el extremo del robot hay que multiplicar por las diferentes matrices de transformación entre el sistema 0 y el sistema 6. Se procedería de la siguiente forma:

**Ecuación 4. Producto para la obtención de la matriz de transformación T.**

$$T = {}^0A_1 \cdot {}^1A_2 \cdot {}^2A_3 \cdot {}^3A_4 \cdot {}^4A_5 \cdot {}^5A_6 = {}^0A_6$$

Como se puede observar en la ecuación 5, la matriz T resultante, está formada por una submatriz 3x3 dedicada a la orientación ( $\vec{n}$ ,  $\vec{o}$ ,  $\vec{a}$ ) y un vector 3x1 dedicado al posicionamiento ( $\vec{p}$ ).

**Ecuación 5. Matriz de Transformación T**

$$T = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & w \end{pmatrix}$$

Para la obtención de los valores de la posición (x, y, z) del extremo del robot nos interesa el exclusivamente la última columna de la matriz de transformación (T). Así, para obtenerlo, realizaremos la siguiente operación:

$${}^0A_6 \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = (x, y, z, w) \rightarrow \vec{p} = (x, y, z)$$

### 5.3. Cinemática Inversa

El objetivo del problema cinemático inverso consiste en encontrar los valores que deben adoptar las coordenadas articulares del robot  $q = [q_1, q_2, q_3, q_4, q_5, q_6]^T$  para que su extremo se posicione y oriente según una determinada localización espacial.

Para resolver este problema encontraremos una solución cerrada que tendrá la siguiente forma:

**Ecuación 6. Forma de las soluciones de la cinemática inversa.**

$$q_k = f_k(x, y, z, \alpha, \beta, \gamma) \text{ para } k = [0, 6]$$

Para el caso que nos ocupa, la solución del problema cinemático inverso no es única, existiendo diferentes soluciones que posicionan y orientan el extremo del robot del mismo modo. No obstante, podemos establecer restricciones para la solución obtenida.

A pesar de las dificultades que se nos plantean se da la circunstancia de que los tres últimos grados de libertad, dedicados fundamentalmente a orientar el extremo del robot, corresponden a giros sobre ejes que se cortan en un punto. De esta forma, plantearemos el problema cinemático inverso para el posicionamiento:

- Posicionamiento: involucra a los valores de los tres primeros ejes ( $q_1, q_2, q_3$ ) y depende del punto del espacio objetivo ( $x, y, z$ ).

**Ecuación 7. Forma de las soluciones de posicionamiento.**

$$q_k = f_k(x, y, z) \text{ para } k = [0, 3]$$

#### 5.3.1. Cinemática inversa para el posicionamiento

Realizamos la resolución del problema cinemático inverso por métodos geométricos. El procedimiento en sí se basa en encontrar suficiente número de relaciones geométricas en las que intervendrán las coordenadas del extremo del robot ( $x, y, z$ ), sus coordenadas articulares ( $q_1, q_2, q_3$ ) y las dimensiones físicas de los eslabones ( $L_1, L_2, L_3$ ).

Aplicamos este método a los primeros 3 GDL de rotación de nuestro robot. Los datos de partida son las coordenadas ( $p_x, p_y, p_z$ ) referidas al sistema de referencia en las que se quiere posicionar su extremo.

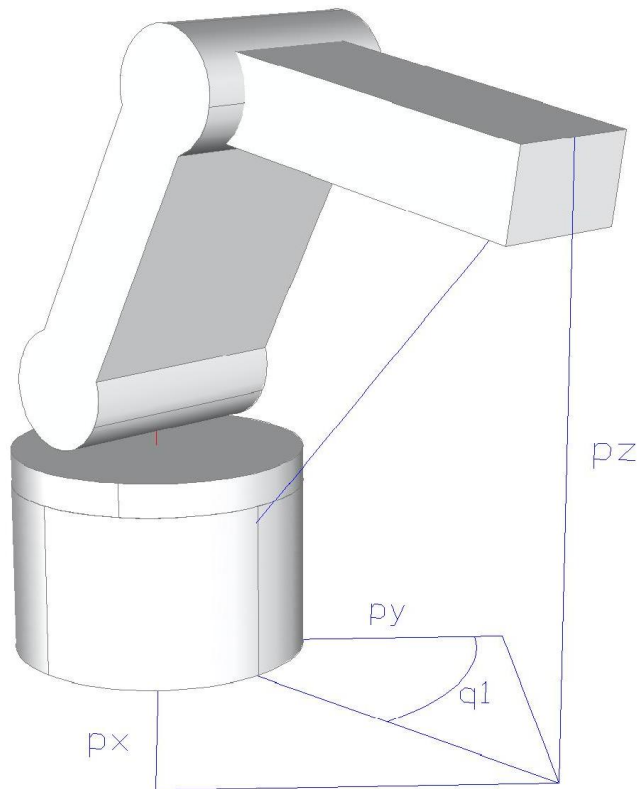


Figura 44 - Esquemático de los 3 primeros GDL del robot.

El valor de  $q_1$  se obtiene de manera inmediata:

$$q_1 = \arctg\left(\frac{p_y}{p_x}\right)$$

Considerando ahora únicamente los eslabones 2 y 3 que están situados en un plano y utilizando el teorema del coseno, se tendrá:

$$\left. \begin{aligned} r^2 &= p_x^2 + p_y^2 \\ r^2 + p_z^2 &= l_2^2 + l_3^2 + 2 l_2 l_3 \cos(q_2) \end{aligned} \right\}$$

$$\cos(q_3) = \frac{p_x^2 + p_y^2 + p_z^2 - l_2^2 - l_3^2}{2 l_2 l_3}$$

Esta expresión permite obtener  $q_3$  en función del vector posición del extremo ( $\vec{p}$ ). No obstante por motivos de ventajas computacionales, es más conveniente utilizar la expresión de la arcotangente en lugar del ascoseno.

Puesto que

$$\cos(q_3) = \pm \sqrt{1 - \cos^2(q_3)}$$

Se tendrá que

$$q_3 = \arctg\left(\frac{\pm \sqrt{1 - \cos^2(q_3)}}{\cos(q_3)}\right)$$

Con

$$\cos(q_3) = \frac{p_x^2 + p_y^2 + p_z^2 - l_2^2 - l_3^2}{2 l_2 l_3}$$

Como se ve, existen dos posibles soluciones para  $q_3$  según tome el signo positivo o negativo en la raíz. Éstas corresponden a las configuraciones de codo arriba y codo abajo del robot.

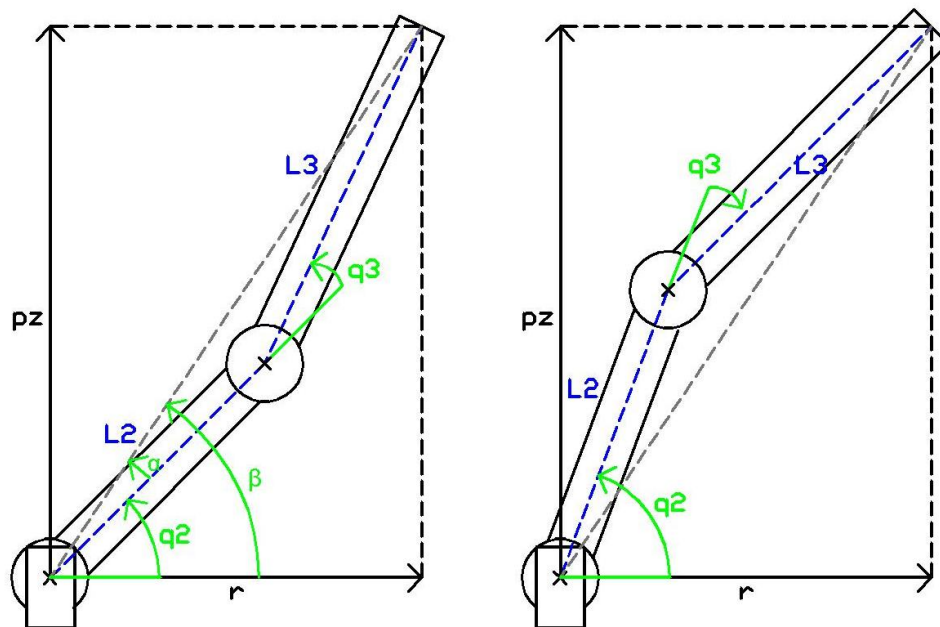


Figura 45 - Configuraciones codo arriba y abajo

El cálculo de  $q_2$  se hace a partir de la diferencia entre  $\beta$  y  $\alpha$ :

$$q_2 = \beta - \alpha$$

Siendo:

$$\beta = \arctg\left(\frac{p_z}{r}\right) = \arctg\left(\frac{p_z}{\pm \sqrt{p_x^2 + p_y^2}}\right)$$

$$\alpha = \arctg\left(\frac{l_3 \sin(q_3)}{l_2 + l_3 \cos(q_3)}\right)$$

Luego, finalmente

$$q_2 = \arctg\left(\frac{p_z}{\pm\sqrt{p_x^2 + p_y^2}}\right) - \arctg\left(\frac{l_3 \operatorname{sen}(q_3)}{l_2 + l_3 \cos(q_3)}\right)$$

De nuevo los dos posibles valores según la elección del signo dan lugar a dos valores diferentes de  $q_2$  correspondientes a las configuraciones codo arriba y abajo.

Las expresiones que resuelven el problema cinemático inverso para los tres primeros grados de libertad del robot son:

$$q_1 = \arctg\left(\frac{p_y}{p_x}\right)$$

$$q_2 = \arctg\left(\frac{p_z}{\pm\sqrt{p_x^2 + p_y^2}}\right) - \arctg\left(\frac{l_3 \operatorname{sen}(q_3)}{l_2 + l_3 \cos(q_3)}\right)$$

$$q_3 = \arctg\left(\frac{\pm\sqrt{1 - \cos^2(q_3)}}{\cos(q_3)}\right)$$



# Capítulo 6

## Conclusiones



## Capítulo 6: Conclusiones

### 6.1. Conclusiones

Mi especialización es la electrónica, emprendí este proyecto con una base pobre en el área de mecánica, sobre todo en la parte de diseño, pero finalmente decidí afrontarlo porque me llamaba mucho la curiosidad, además de verlo como un reto para la nueva adquisición de conocimientos, formarme aún más como ingeniero en diferentes áreas (mecánica por ejemplo), y francamente no me ha defraudado en absoluto, no sólo he ampliado mis conocimientos en otros campos, sino que me ha abierto las puertas a un mundo que desconocía totalmente, en el que se mezcla la electrónica y la mecánica... la mecatrónica. Algo parecido me pasaba con el mundo de las impresoras 3D, el cual desconocía por completo y que ha logrado engancharme hasta el punto de tener como próximo objetivo construir una impresora 3D para continuar con mis propios proyectos y con mi formación.

He aprendido a verme capacitado, con los problemas que me han ido surgiendo, a solventar de manera autónoma casi cualquier tipo de problema que se me plantee a la hora de abordar un proyecto de estas características.

En lo que respecta al brazo robótico, se han logrado cumplir los objetivos de forma satisfactoria, a pesar de todos los problemas que ha ido apareciendo a lo largo de su realización.

Todos y cada uno de los problemas surgidos me han dado amplitud de conocimientos, pues me han hecho investigar y adentrarme más de lleno en el problema. Por ejemplo, el primer problema que me surgió fue porque el eslabón dos del brazo fue diseñado para moverse con dos servomotores S3003 enfrentados (solución más barata). Una vez que se implementó y se probó el movimiento del eslabón con un sencillo programa realizado en Arduino, se observó que los servos se sobrecalentaban demasiado debido a que aun siendo el mismo servo, cada uno posee su propio error en el movimiento (aproximadamente un grado) y trataba constantemente de autoreajustarse. La primera idea fue cambiar los servos por uno sólo que haga la misma función, pero encarecía el proyecto y se trató de solucionar por software probando diferentes funciones. Se consiguió evitar el sobrecalentamiento, en parte, pero aún se seguían sobrecalentando demasiado, teniendo finalmente que poner el servo S3010. Con este servo todo iba perfecto hasta que se montó el brazo completo. En este momento el servomotor se quedaba sin fuerzas para levantar el brazo cuando se encontraba extendido del todo sobre la horizontal, obligándome a poner un servo mayor más caro, el Futaba S3305, con el que no ha vuelto a dar problemas. Con este problema aprendí funciones de programación en Arduino que desconocía e incluso investigué el tipo de control que tenían los servos. Comprendí las diferencias entre servos analógicos y digitales. En definitiva, me hizo crecer como ingeniero en terrenos que desconocía.

Por otro lado, en relación con el párrafo anterior, la pinza ha tenido que ser tratada por software de manera especial. El control de la pinza se ha creado de manera que se autoregule, es decir, coge la pieza, lee su posición y se envía la señal correspondiente a la apertura de la pinza adaptándose a la pieza cogida. Durante las pruebas realizadas, a la pinza simplemente se enviaba 0° (abierta) ó 180° (cerrada). Esto conllevaba a un calentamiento del servo innecesario, porque si la pinza con la pieza cogida mantenía un ángulo de 60° y se le enviaba 180°, la señal que enviaba al motor el control del servo era máxima, tratando de llevarlo a su posición correcta provocando el calentamiento. Esta autoregulación puede valer también para detectar si ha

cogido o no pieza, simplemente enviando la señal de cierre y leyendo su posición nos bastaría para saberlo, pues si la posición son  $180^\circ$  está completamente cerrada y por lo tanto no hay pieza en ella.

Otro problema que se me planteó y me llevó bastante tiempo solucionar fue a la hora de gestionar el control en posición, pues como se ha comentado en el capítulo tres, la señal continua del potenciómetro interno de los servos, tenía una componente residual que impedía su lectura de forma precisa, haciendo que el conversor AD del microcontrolador variase su lectura para una misma posición. La solución fue crear varios tipos de lecturas (cuyos flujogramas se encuentran en el capítulo tres) leyendo varias veces la posición y tratando de tener un valor medio de dichas lecturas, al repetir el proceso varias veces y comparar las medias, se obtiene finalmente un valor más preciso (difiere en 3 grados máximo la lectura estimada de la enviada por la PWM). Aún teniendo esta solución, traté de eliminar la señal mediante electrónica adicional con la adición de filtros, condensadores, diodos... y cuando conseguía estabilizar la señal, el servo oscilaba y no se mantenía en su posición, se desestabilizaba en cada movimiento, pues el rizado variaba en función de la posición del potenciómetro. Finalmente se dejó la solución de la estimación por software, ya que el brazo no requiere de gran precisión para la aplicación que ha sido diseñado (uso educativo). Por lo tanto, la precisión de 3 grados máximo de error en la medida, se puede considerar buena para aplicaciones de este tipo. Aún así la repetitividad del brazo en ir a un punto con una determinada configuración es bastante buena, (como se puede ver en el video demostrativo cuyo enlace está en el resumen de este proyecto), la caja que mueve primero tiene poco margen de error entre la pinza y las paredes pues casi son de la misma anchura y la coge sin ningún problema.

A la vista de todo esto, el control en posición se encuentra realizado, el problema reside en el control en velocidad. Este conflicto viene generado por el uso de servomotores controlados en posición, pues la velocidad de estos no es algo que se pueda controlar fácilmente ya que depende del peso que estén soportando, la fuerza que tengan que realizar, etc. Por ejemplo, se trata de servos que reaccionan muy rápidamente y para el brazo sobre todo, en el eslabón dos, cuando está completamente extendido, es un grave problema, pues debido a las holguras con las piezas y a las holguras que meten los servos, cuando se traslada el brazo entre posiciones alejadas, puede provocar la rotura del servo, oscilando demasiado al llegar a la posición hasta que se coloca. Pega un golpe seco y puede romper los engranajes. La solución que se ha implementado es frenar el motor por software cuando está llegando a la posición deseada, evitando así que llegue a máxima velocidad a la posición, generando oscilaciones que puede provocar que el extremo del robot choque contra algo. Ésta es una solución al problema pero no es un control en velocidad propiamente dicho. Uno de los grandes inconvenientes de esta solución es que mientras se mueve el eslabón dos, el resto de eslabones permanecen inmóviles hasta acabar el movimiento. Quizás entrando un poco más en las características del microcontrolador y gestionando las interrupciones junto con los timers, se pudiera conseguir el control en velocidad buscado, pero ha sido imposible realizarlo de momento antes de la entrega del proyecto por falta de tiempo, al menos el intento.

Un último problema a mencionar y que no se ha podido solucionar, es el giro completo de la base del robot. La base se compone de dos servos que giran  $180^\circ$  cada uno, por lo que debería girar  $360^\circ$  y no es el caso debido a la cantidad de cables que hay que llevar desde los servos hasta la placa. En la estructura, internamente, generan una serie de rozamientos provocando que en vez de girar  $360^\circ$  gire  $330^\circ$  aproximadamente.

En lo que respecta a posibles mejoras, los cuatro últimos problemas mencionados (precisión del control en posición, oscilaciones en posiciones alejadas, control en velocidad y giro completo de la base) podrían solventarse si se evitara poner los servos en las articulaciones y se colocaran en la base, utilizando en su lugar para generar el movimiento, un sistema de transmisión junto con un potenciómetro para la lectura de la posición en cada articulación. Además de cambiar los servomotores por motores de continua, de manera que, permitiese un mejor control tanto en posición como en velocidad, sobre todo en este último, que es el más difícil de conseguir con servos.

No todo son mejoras futuras, pues durante la etapa de diseño y pruebas de funcionamiento, también se han realizado mejoras en el brazo, por ejemplo, el diseño inicial de la muñeca del robot, no permitía aplicar el desacoplo cinemático, porque los ejes de las articulaciones no se cortaban en ningún punto en común, complicando mucho el desarrollo del modelo. Finalmente se modificaron las piezas necesarias consiguiendo eliminar esa complicación.

Para terminar, como conclusión y reflexión personal, he de mencionar que la realización de este proyecto me ha ayudado a poner en práctica la mayor parte de los conocimientos adquiridos a lo largo de la carrera, requiriendo de asignaturas que no creí que fuera a utilizar, por carecer de sentido práctico en el momento en que se cursaron. Se podría decir que me ha abierto los ojos y ha conseguido eliminar esa idea que se suele quedar en la cabeza cuando se terminan los estudios, la idea de creer tener las herramientas necesarias para hacer algo pero no saber cómo usarlas. En mi caso he aprendido a usarlas, encontrándome con ánimos de emprender nuevos proyectos.

Finalmente a modo de resumen, se podría decir que este proyecto se basaba principalmente en comprobar la viabilidad de hacer un brazo robótico imprimible educacional, que tuviera control en posición y fuera manejado por puerto serie. Una vez que se ha visto que es viable tras cumplir los objetivos, han empezado a crearse proyectos de mejora, como es el caso del proyecto de J. Carlos Rodríguez Zambrana, quien está creando el control del mismo, permitiendo controlarlo mediante gamepad, junto con un simulador que permite el control con varios modos de funcionamiento. Siendo ambos proyectos de licencia libre. En definitiva, se trata de un proyecto aún en desarrollo y del que, en un futuro, mejora tras mejora, se puede obtener un buen diseño, con infinitud de posibilidades, y no nos olvidemos, a un bajo coste.



# REFERENCIAS

- Antonio Barrientos, Luis Felipe Peñín, Carlos Balaguer y Rafael Arecil: “**Fundamentos de Robótica Industrial**” Ed. McGraw-Hill 1997.
- Pagina web de OpenScad, uso del manual: <http://www.openscad.org> [Octubre 2011]
- Web Futaba para comparar servos : <http://www.futaba-rc.com/servos/analog.html> [Noviembre 2011]
- Sitio web makerbot información sobre la impresora a utilizar: <http://wiki.makerbot.com/start> [Noviembre 2011]
- Información sobre servomotores, cómo funcionan: [http://cfievalladolid2.net/tecno/cyr\\_01/robotica/sistema/motores\\_servo.htm](http://cfievalladolid2.net/tecno/cyr_01/robotica/sistema/motores_servo.htm) [Diciembre 2011]
- Sitio web Arduino dudas sobre programación, sobre la placa: <http://www.arduino.cc/es/> [Octubre 2011-Mayo 2012]
- Manuales iearobotics, modificación de futaba s3003: [http://www.iearobotics.com/wiki/index.php?title=P%C3%A1gina\\_Principal](http://www.iearobotics.com/wiki/index.php?title=P%C3%A1gina_Principal) [Enero 2012, Junio 2012]
- Ejemplos puerto serie Arduino, usar teclado para comunicarse: <http://exergy123.blogspot.com.es/search/label/Arduino> [Abril 2012]
- Web robotica ABB tipos de brazos roboticos : <http://www.abb.es/> [Marzo 2012]
- Web de KUKA, diseños de robots <http://www.kuka-robotics.com/es/> [Marzo 2012]
- Arduino reference para dudas: <http://arduino.cc/en/Reference/HomePage> [Julio 2012]
- Uso de las impresoras: <http://replicat.org/> [Diciembre 2011]
- Búsqueda de la pinza, objetos para el agarre, etc : <http://www.thingiverse.com/> [Noviembre 2011- Agosto 2012]



# **ANEXO I**

## **Tiempo de impresión y presupuesto**

### A. Tiempo de impresión

A continuación se exponen el tiempo empleado para la impresión de cada pieza del eslabón, el tiempo de impresión de cada eslabón y el tiempo de impresión empleado en total.

#### - Eslabón 0

Número de copias	Pieza	Tiempo de impresión (minutos)
1	A1	42
1	A2	35
1	B	72

Tabla 11 - ANEXO I/A. Tiempo de impresión piezas Eslabón 0

#### - Eslabón 1

Número de copias	Pieza	Tiempo de impresión (minutos)
1	A	65
1	B	42
1	C	26
1	D	9

Tabla 12 - ANEXO I/A. Tiempo de impresión piezas Eslabón 1

#### - Eslabón 2

Número de copias	Pieza	Tiempo de impresión (minutos)
2	A1	27
2	A2	26
2	B	11
1	C1	30
1	C2	28
1	D	34
1	E	13

Tabla 13 - ANEXO I/A. Tiempo de impresión piezas Eslabón 2



---

- Eslabón 3

Número de copias	Pieza	Tiempo de impresión (minutos)
1	A	27
1	B	25
2	C	18
2	D	8
1	E1	19
1	E2	16
1	F	24

Tabla 14 - ANEXO I/A. Tiempo de impresión piezas Eslabón 3

---

- Eslabón 4

Número de copias	Pieza	Tiempo de impresión (minutos)
1	A	21
1	B	16
1	C	10

Tabla 15 - ANEXO I/A. Tiempo de impresión piezas Eslabón 4

---

- Eslabón 5

Número de copias	Pieza	Tiempo de impresión (minutos)
1	A	23
1	B	12
1	C	12

Tabla 16 - ANEXO I/A. Tiempo de impresión piezas Eslabón 5

---

- Pinza

Número de copias	Pieza	Tiempo de impresión (minutos)
1	ENGRANAJES	18
1	RESTO DE PIEZAS	76

Tabla 17 - ANEXO I/A. Tiempo de impresión piezas Pinza

Tiempo por eslabón y total:

Eslabón	Tiempo
0	149
1	142
2	169
3	137
4	47
5	47
Pinza	94
<b>Tiempo total del brazo robótico</b>	<b>785 minutos (13.08 horas)</b>

Tabla 18 - ANEXO I/A. Tiempo de impresión por eslabones y tiempo de impresión de A-Bot completo.

## B. Presupuesto

Cantidad	Artículo	Precio (€ /unidad)	Total €
3	Futaba S3003	8.50	25.50
1	Futaba S3305	41	41.00
3	Tower pro MG90S	9.50	28.50
1	Hitec HS—55	8.90	8.90
2	Cable tririple plano ( 5m paquete)	3.9	7.80
1	Cable bipolar 0.7mm <sup>2</sup> (5m)	0.30	0.30
1	Bridas	0.60	0.60
2	Manguera recoge cables	3.50	7.00
32	Tornillo helicoidal 5x15mm (rosca 2,5mm)	0.25	8.00
8	Tornillo de métrica M3x25mm	0.75	6.00
8	Tuerca de métrica M3	0.20	1.60
2	Tornillo de métrica M2x6mm (20 unidades)	3.30	6.60
3	Tira 50 circuitos machos pcb	0.99	2.97
1	Tira 40 pines hembra pcb	0.99	0.99
1	Zócalo 16 pines	0.10	0.10
2	Regleta 2 contactos R =5 PCB	0.19	0.38
1	Placa fibra de vidrio positiva 60x80	1.31	1.31
1	Arduino UNO	28.91	28.91
1	74HC4052 - CMOS 2X4-Channel analog MUX	0.16	0.16
1	Tubo termoretractil	0.50	0.50
0.06	Plástico ABS	50	3.00
<b>Total</b>			<b>177.138 €</b>

Tabla 19 - ANEXO I/B. Presupuesto prototipo A-Bot completo.



# **ANEXO II**

## **Proyecto de J. Carlos Rodríguez Zambrana (Breve resumen. Objetivos)**

## Proyecto de J. Carlos Rodríguez Zambrana (Breve resumen. Objetivos)

El presente proyecto, se ha basado fundamentalmente en el diseño total del brazo robótico orientado para docencia, y en su gobierno mediante comandos enviados a través del puerto serie, permitiendo el control de posición en bucle cerrado y su movimiento.

Tras haber realizado el diseño y vista la viabilidad del mismo, el departamento de Ingeniería de Sistemas y Automática ha lanzado paralelamente otro proyecto cuyo objetivo principal es crear el modelo cinemático y control de un brazo robótico imprimible cuyo autor es J. Carlos Rodríguez Zambrana, donde utilizó el prototipo A-Bot (anteriormente expuesto).

En dicho proyecto se ha implementado y diseñado un programa en C++ y OpenGL. Donde lo primero que llama la atención es su interfaz gráfica, en la que se representa una estructura similar a la del A-Bot con todas y cada una de sus características y movimientos relativos, como se puede observar en la figura 46, en las cuales aparecen como se han ido dibujando los diferentes eslabones hasta construir el brazo robótico completo (véase figura 47):

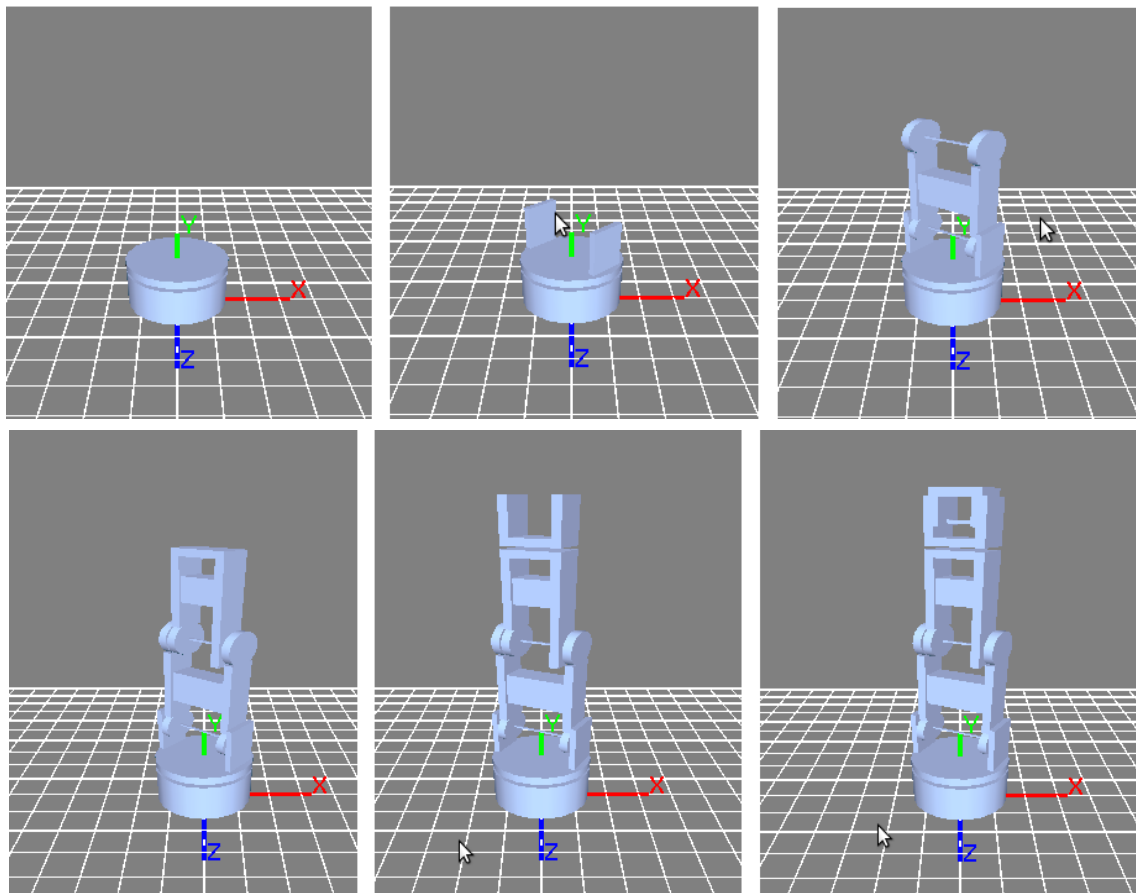


Figura 46 - ANEXO II. Secuencia de dibujado de los eslabones

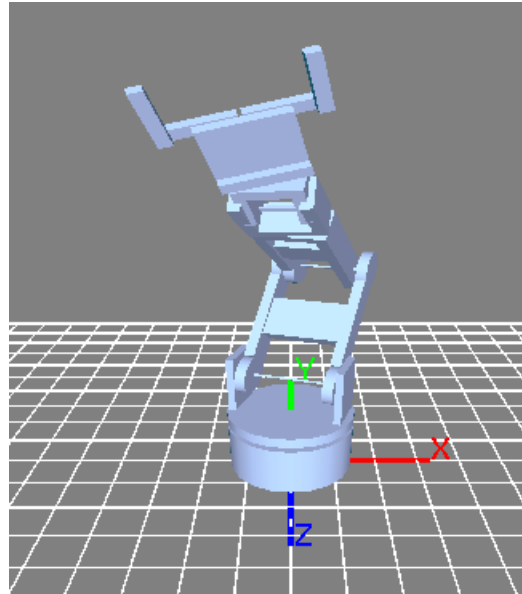


Figura 47 - ANEXO II. Representación del robot completo con la pinza abierta en su extremo.

Otra de las principales características que tiene el simulador es que en su pantalla, se puede observar en todo momento tanto el valor de los ángulos de los ejes como la posición en la que se encuentra el extremo del robot (véase figura 48).

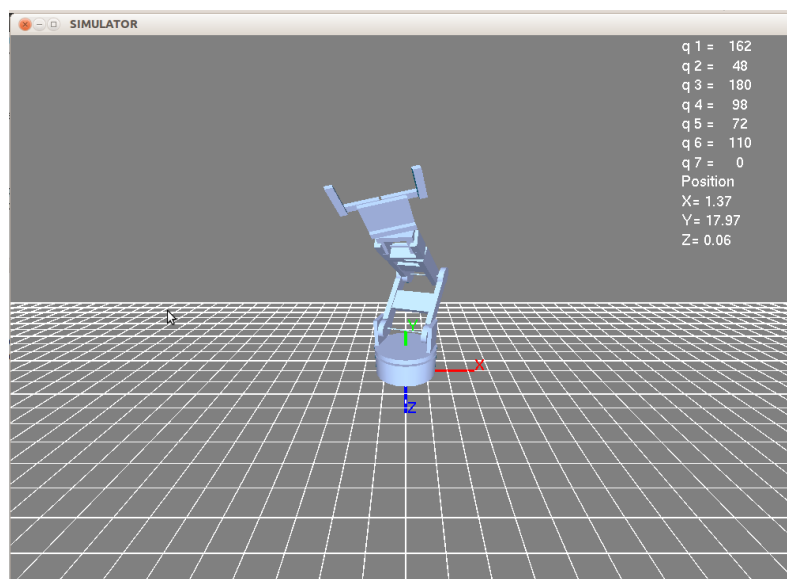


Figura 48 - ANEXO II. Captura de pantalla del simulador.

Todo esto es a grandes rasgos lo referente a la parte visual del simulador, en cuanto a su manejo se ha diseñado con tres modos de funcionamiento:

- Control mediante gamepad, donde el usuario mueve el brazo simulado transfiriendo los movimientos al brazo real posteriormente.
- Modo de grabación, donde el simulador permite la grabación de una secuencia de movimientos que posteriormente serán transferidos al prototipo para su ejecución.

- Modo de posicionamiento, este modo de funcionamiento permite al usuario, llevar el extremo del robot a las coordenadas de un punto que haya introducido con anterioridad.

El sistema de conexionado, que se plantea en el proyecto, entre el PC y A-Bot, puede ser de dos tipos, bien por cable USB, donde al ir por cable se trata de una conexión más segura, ó bien por Bluetooth, que permite el control desde el ordenador sin necesidad de estar conectado físicamente al robot, pero con la limitación de que no se puede sobrepasar los nueve metros entre ambos o la conexión podría perderse.

Finalmente al igual que ocurre con el brazo robótico diseñado A-Bot, el software creado para su control se trata de código abierto por lo tanto está a disposición de cualquier usuario, con la finalidad de poder modificarlo sin problema alguno, pudiendo ampliarlo, o mejorarlo.